

SAK-CIC310-OSMX2HT

FlexRay Communication Controller

IFLEX

Microcontrollers



Never stop thinking

Edition 2007-02

**Published by
Infineon Technologies AG
81726 München, Germany**

**© Infineon Technologies AG 2008.
All Rights Reserved.**

Legal Disclaimer

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie"). With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (www.infineon.com).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

SAK-CIC310-OSMX2HT

FlexRay Communication Controller

IFLEX

Step A11

Microcontrollers



Never stop thinking

TriCore™ is a trademark of Infineon Technologies AG

FlexRay™ and the FlexRay logo  are registered trademarks of the FlexRay Consortium.

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all? Your feedback will help us to continuously improve the quality of this document.

Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com



Table of Contents

This "Table of Contents" section refers to page numbers in the SAK-CIC310-OSMX2HT User's Manual.

1	Architecture Overview	1-1
1.1	About this Document	1-1
1.1.1	Related Documentations	1-1
1.1.2	Text Conventions	1-1
1.1.3	Reserved, Undefined, and Unimplemented Terminology	1-3
1.1.4	Register Access Modes	1-5
1.1.5	Abbreviations	1-6
1.2	Summary of Features	1-9
1.3	System Integration Concept	1-11
1.3.1	MLI Host Link (Option One)	1-11
1.3.2	SSC Host Link (Option Two)	1-12
1.3.3	Parallel Host Link (Option Three)	1-14
1.3.4	DMA Controller	1-14
1.4	General Device Information	1-17
1.4.1	Block Diagram	1-17
1.4.2	Pin Definition and Functions	1-19
1.4.2.1	Package Outline	1-19
1.4.2.2	Pin Description	1-21
1.4.2.3	Ordering Information	1-33
2	FlexRay Protocol Controller (E-Ray)	2-1
2.1	Overview	2-1
2.2	Definitions	2-2
2.3	Block Diagram	2-2
2.4	Programmer's Model	2-5
2.4.1	Register Map	2-5
2.4.2	E-Ray Kernel Registers	2-6
2.4.2.1	Customer Registers	2-11
2.4.2.2	Special Registers	2-18
2.4.2.3	Service Request Registers	2-28
2.4.2.4	Communication Controller Control Registers	2-71
2.4.2.5	Communication Controller Status Registers	2-99
2.4.2.6	Message Buffer Control Registers	2-121
2.4.2.7	Message Buffer Status Registers	2-128
2.4.2.8	Identification Registers	2-148
2.4.2.9	Input Buffer	2-150
2.4.2.10	Output Buffer	2-161

2.5	Functional Description	2-176
2.5.1	Communication Cycle	2-176
2.5.1.1	Static Segment	2-176
2.5.1.2	Dynamic Segment	2-177
2.5.1.3	Symbol Window	2-177
2.5.1.4	Network Idle Time (NIT)	2-177
2.5.1.5	Configuration of Network Idle Time (NIT) Start and Offset Correction Start.	2-177
2.5.2	Communication Modes	2-179
2.5.3	Clock Synchronization	2-179
2.5.3.1	Global Time	2-179
2.5.3.2	Local Time	2-179
2.5.3.3	Synchronization Process	2-180
2.5.3.4	External Clock Synchronization	2-182
2.5.4	Error Handling	2-182
2.5.4.1	Clock Correction Failed Counter	2-183
2.5.4.2	Passive to Active Counter	2-184
2.5.4.3	HALT Command	2-184
2.5.4.4	FREEZE Command	2-184
2.5.5	Communication Controller States	2-185
2.5.5.1	Communication Controller State Diagram	2-185
2.5.5.2	DEFAULT_CONFIG State	2-187
2.5.5.3	MONITOR_MODE	2-188
2.5.5.4	READY State	2-188
2.5.5.5	WAKEUP State	2-189
2.5.5.6	STARTUP State	2-193
2.5.5.7	Startup Timeouts	2-196
2.5.5.8	Path of leading Coldstart Node (initiating coldstart)	2-197
2.5.5.9	NORMAL_ACTIVE State	2-199
2.5.5.10	NORMAL_PASSIVE State	2-199
2.5.5.11	HALT State	2-200
2.5.6	Network Management	2-201
2.5.7	Filtering and Masking	2-201
2.5.7.1	Frame ID Filtering	2-202
2.5.7.2	Channel ID Filtering	2-202
2.5.7.3	Cycle Counter Filtering	2-203
2.5.7.4	FIFO Filtering	2-204
2.5.8	Transmit Process	2-205
2.5.8.1	Static Segment	2-205
2.5.8.2	Dynamic Segment	2-205
2.5.8.3	Transmit Buffers	2-205
2.5.8.4	Frame Transmission	2-207
2.5.8.5	Null Frame Transmission	2-207

2.5.9	Receive Process	2-207
2.5.9.1	Frame Reception	2-208
2.5.9.2	Null Frame reception	2-208
2.5.10	FIFO Function	2-209
2.5.10.1	Description	2-209
2.5.10.2	Configuration of the FIFO	2-210
2.5.10.3	Access to the FIFO	2-211
2.5.11	Message Handling	2-211
2.5.11.1	Host access to Message RAM	2-211
2.5.11.2	Data Transfers between IBF / OBF and Message RAM	2-217
2.5.11.3	Minimum eray_bclk	2-223
2.5.11.4	FlexRay Protocol Controller access to Message RAM	2-227
2.5.12	Message RAM	2-228
2.5.12.1	Header Partition	2-230
2.5.12.2	Data Partition	2-233
2.5.12.3	Parity Check	2-234
2.6	Module Service Request	2-237
2.7	Restrictions	2-240
2.7.1	Message Buffers with the same Frame ID	2-240
2.7.2	Data Transfers between IBF / OBF and Message RAM	2-240
2.8	Known non functional Features of E-Ray Module Revision 1.0.0	2-241
2.9	Revision History	2-251
3	System Control Unit (SCU)	3-1
3.1	SCU Registers	3-1
3.2	Reset Control Block	3-4
3.2.1	Reset Length	3-4
3.3	Clock System	3-5
3.3.1	Overview	3-5
3.3.2	Clock Generation Unit	3-5
3.3.2.1	Crystal Oscillator Circuit (COSC)	3-6
3.3.3	Phase-Locked Loop (PLL) Module	3-13
3.3.3.1	PLL Functional Description	3-15
3.3.3.2	Loss of Oscillator Lock Operation	3-27
3.3.3.3	Loss of Oscillator Lock Recovery	3-28
3.3.4	Clock Multiplex Unit	3-28
3.4	Power Supply System	3-30
3.4.1	Miscellaneous SCU Registers	3-31
3.5	Interrupt Control	3-33
3.5.1	Internal Interconnection Signals	3-34
3.5.2	External Trigger Inputs	3-34
3.5.3	Interrupt Output Structure	3-38
3.5.3.1	Communication Relations	3-38

3.5.3.2	Service Request Routing	3-39
3.5.4	DMA Channel Trigger	3-43
3.6	Mode Selection	3-46
3.6.1	MODE 0 and Mode 1 Pin	3-46
3.6.2	JTAGEN Pin	3-46
3.6.3	XMU Global Control Register	3-47
3.6.4	DMA Reprogramming	3-49
3.6.5	Gather Scattered Bits	3-51
3.6.6	Address Map	3-55
4	Host Communication Channels	4-1
4.1	Host Write Request	4-1
4.1.1	MLI Write Operation	4-1
4.1.2	SSC Write Operation	4-2
4.1.3	XMU Write Operation	4-2
4.2	Host Read Request	4-2
4.2.1	MLI Read Operation	4-2
4.2.2	SSC Read Operation	4-3
4.2.3	XMU Read Operation	4-3
4.3	SSC Move Engine	4-4
4.4	Communication Principles MLI	4-8
4.4.1	Prerequisites for Communication	4-9
4.4.2	Enable MLI Interfaces	4-9
4.4.3	Single steps of Communication via MLI bridge	4-10
4.4.4	DMA Support for MLI (Single Message Buffer Read Support)	4-10
4.4.4.1	Single Message Buffer Read Support	4-10
4.4.4.2	Periodically Copy Message Buffer Contents to Host Memory	4-21
4.5	Communication Principles SSC	4-39
4.5.1	Single Read Access via SSC	4-45
4.5.2	Consecutive Read Accesses via SSC	4-47
4.5.3	Single Write Access via SSC	4-51
4.5.4	Consecutive Write Access via SSC	4-53
4.5.5	Error Handling	4-57
5	Direct Memory Access Controller (DMA)	5-1
5.1	DMA Controller Description	5-1
5.1.1	Features	5-1
5.1.2	Definition of Terms	5-2
5.1.3	DMA Principle	5-3
5.1.4	DMA Block Diagram	5-4
5.1.5	DMA Operation Functionality	5-4
5.1.5.1	Shadow Registers	5-5
5.1.5.2	DMA Channel Request Control	5-8
5.1.5.3	DMA Channel Operation Mode	5-8

5.1.5.4	Move Count	5-12
5.1.5.5	Request Lost	5-12
5.1.5.6	Circular Buffer	5-13
5.1.5.7	Interrupt Signal Generation	5-14
5.1.5.8	Pattern Detection	5-14
5.1.5.9	Error Conditions	5-16
5.1.5.10	Channel Reset Operation	5-16
5.1.5.11	Programmable Address Modification	5-17
5.1.6	Transaction Control Engine	5-19
5.1.7	Request Assignment Unit	5-20
5.1.8	General Interrupt Structure	5-21
5.2	DMA Module Kernel Registers	5-22
5.2.1	Overview	5-22
5.2.2	Identification Register	5-25
5.2.2.1	DMA Module Identification Register	5-25
5.2.2.2	Global Interrupt Set Register	5-26
5.2.3	General Control and Status Registers	5-27
5.2.3.1	Channel Reset Request Register	5-27
5.2.3.2	Transaction Request State Register	5-28
5.2.3.3	Software Transaction Request Register	5-29
5.2.3.4	Hardware Transaction Request Register	5-30
5.2.3.5	Enable Error Register	5-31
5.2.3.6	Error Status Register	5-33
5.2.3.7	Clear Error Register	5-35
5.2.3.8	Interrupt Status Register	5-37
5.2.3.9	Wrap Status Register	5-38
5.2.3.10	Interrupt Clear Register	5-39
5.2.4	Move Engine Registers	5-40
5.2.4.1	Move Engine Status Register	5-40
5.2.4.2	Move Engine 0 Read Register	5-42
5.2.4.3	Move Engine 0 Pattern Register	5-43
5.2.5	Channel Control, Status and Address Registers	5-45
5.2.5.1	Channel 0n Control Register (n = 0-7)	5-45
5.2.5.2	Channel 0n Status Register (n = 0-7)	5-49
5.2.5.3	Channel Interrupt Control Register (n = 0-7)	5-50
5.2.5.4	Address Control Register (n = 0-7)	5-52
5.2.5.5	Source Address Register (n = 0-7)	5-55
5.2.5.6	Destination Address Register (n = 0-7)	5-56
5.2.5.7	Shadow Address Register (n = 0-7)	5-57
5.3	DMA Module Implementation	5-58
5.3.1	DMA Request Assignment Matrix	5-58
5.3.2	Address Map	5-61

6	Synchronous Serial Interface (SSC)	6-1
6.1	SSC Kernel Description	6-1
6.1.1	Overview	6-1
6.1.2	General Operation	6-2
6.1.2.1	Operating Mode Selection	6-3
6.1.2.2	Full-Duplex Operation	6-5
6.1.2.3	Half-Duplex Operation	6-7
6.1.2.4	Continuous Transfers	6-9
6.1.2.5	Port Control	6-9
6.1.2.6	Baud Rate Generation	6-10
6.1.2.7	Slave Select Input Operation	6-12
6.1.2.8	Slave Select Output Generation Unit	6-13
6.1.2.9	Shift Clock Generation	6-16
6.1.2.10	Error Detection Mechanisms	6-16
6.2	SSC Kernel Registers	6-19
6.3	SSC Module Implementation	6-35
6.3.1	Interfaces of the SSC Module	6-35
6.3.1.1	Port Connections of SSC	6-35
6.3.1.2	Interface signals of SSC	6-35
6.3.1.3	RDY signal	6-35
6.3.1.4	DIR Signal	6-36
6.3.1.5	Connecting 2 or more slaves to 1 host	6-36
6.3.1.6	Half-Duplex Control	6-37
6.3.2	Error Handling	6-38
6.3.3	Reset Initialization	6-40
6.3.4	Address Map	6-41
7	Micro Link Serial Bus Interface (MLI)	7-1
7.1	MLI Applications	7-1
7.2	MLI Kernel Description	7-3
7.2.1	Overview	7-3
7.2.1.1	Naming Conventions	7-4
7.2.1.2	MLI Communication Principles	7-5
7.2.2	General Description	7-6
7.2.3	Handshake Description	7-13
7.2.4	Startup Procedure	7-15
7.2.5	MLI Kernel and MLI Interface Logical Connection	7-16
7.2.6	MLI Transmitter	7-17
7.2.6.1	MLI Transmitter Reset	7-17
7.2.6.2	MLI Transmitter Operation Modes	7-17
7.2.6.3	Internal Architecture And Interface Signals	7-19
7.2.6.4	Transmission Format	7-20
7.2.6.5	Transmission Modes	7-21

7.2.6.6	Transfer Mode Selection	7-28
7.2.6.7	Parity Generation	7-31
7.2.6.8	Error Detection and Handling	7-31
7.2.6.9	MLI Transmitter Input/Output Control	7-33
7.2.7	MLI Receiver	7-34
7.2.7.1	MLI Receiver Reset	7-34
7.2.7.2	MLI Receiver Operation Modes	7-34
7.2.7.3	Internal Architecture And Interface Signals	7-36
7.2.7.4	MLI Receiver Operation	7-36
7.2.7.5	Error Handling	7-42
7.2.7.6	MLI Receiver Input/Output Control	7-44
7.2.8	Reading Process Summary	7-46
7.2.9	MLI Interrupts	7-47
7.2.10	Clock Domains and Handshake Timing	7-50
7.2.11	Data Flow Description	7-54
7.2.11.1	Copy Base Address	7-54
7.2.11.2	Command Frame	7-55
7.2.11.3	Write Frame	7-56
7.2.11.4	Read Frame	7-57
7.2.11.5	Access to Remote Window	7-59
7.2.12	Baud Rate Generation	7-60
7.2.12.1	Fractional Divider	7-61
7.3	MLI Kernel Registers	7-64
7.3.1	Module Identification Register	7-67
7.3.2	MLI Transmitter Registers	7-68
7.3.2.1	Transmitter Control Register	7-68
7.3.2.2	Transmitter Status Register	7-71
7.3.2.3	Transmitter Pipe Status Registers	7-73
7.3.2.4	Transmitter Command Register	7-75
7.3.2.5	Transmitter Registers Status Register	7-77
7.3.2.6	Transmitter Pipe Address Offset Registers	7-79
7.3.2.7	Transmitter Pipe Data Registers	7-80
7.3.2.8	Transmitter Data Read Answer Register	7-81
7.3.2.9	Transmitter Pipe Base Address Registers	7-82
7.3.2.10	Transmitter Copy Base Address Register	7-83
7.3.3	MLI Receiver Registers	7-84
7.3.3.1	Receiver Control Register	7-84
7.3.3.2	Receiver Pipe Base Address Registers	7-87
7.3.3.3	Receiver Pipe Status Registers	7-88
7.3.3.4	Receiver Address Register	7-89
7.3.4	Receiver Data Register	7-90
7.3.5	MLI Kernel Common Registers	7-91
7.3.5.1	Set Clear Register	7-91

7.3.6	MLI Interrupt Registers	7-93
7.3.6.1	Transmitter Interrupt Enable Register	7-93
7.3.6.2	Transmitter Interrupt Status Register	7-95
7.3.6.3	Transmitter Interrupt Node Pointer Register	7-96
7.3.6.4	Receiver Interrupt Enable Register	7-98
7.3.6.5	Receiver Interrupt Status Register	7-100
7.3.6.6	Receiver Interrupt Node Pointer Register	7-101
7.3.6.7	Global Interrupt Set Register	7-103
7.3.6.8	Output Input Control Register	7-104
7.3.6.9	Fractional Divider Registers	7-109
7.4	MLI Module Implementation	7-111
7.4.1	Interfaces of the MLI Module	7-111
7.4.1.1	Port Connections of MLI	7-111
7.4.1.2	Interface signals of MLI	7-111
7.4.2	Address Map	7-112
8	External Memory Interface Unit (XMU)	8-1
8.1	Overview	8-1
8.2	XMU Features	8-1
8.3	Basic XMU Operation	8-2
8.4	XMU Signal Description	8-3
8.4.1	Address Bus, A[12:0]	8-3
8.4.2	Data Bus, D[16:0]	8-3
8.4.2.1	Data Bus Direction Control	8-4
8.4.3	Read and Write Strokes, \overline{RD} and \overline{WR}	8-4
8.4.4	Byte Control Signals, \overline{BCx}	8-4
8.4.5	Variable Wait State Control, \overline{WAIT}	8-4
8.4.6	XMU Chip Select, \overline{CSFPI}	8-5
8.5	Detailed External to Internal XMU Operation	8-5
8.5.1	XMU Signal Direction	8-5
8.5.2	Address Translation	8-6
8.5.3	External to Internal Access Controls	8-9
8.5.4	Basic Access Timing	8-9
8.5.5	External Requests to the XMU	8-12
8.5.6	Bidirectional Data Lines	8-12
8.6	XMU Registers	8-13
8.6.1	Global Control Register	8-13
8.6.2	Byte Control Select	8-14
8.7	XMU Module Implementation	8-16
8.7.1	Interfaces of the XMU Module	8-16
8.7.2	External Address Extension	8-16
8.7.3	External Access Configuration Register	8-19
8.8	Supported Address Range	8-21

9	Parallel Ports (PORTS)	9-1
9.1	General Description	9-1
9.1.1	Input Stage Control	9-1
9.1.1.1	Reset Behavior	9-2
9.1.2	Output Driver Control	9-2
9.1.2.1	Reset Behavior	9-2
9.1.2.2	Power-fail Behavior	9-2
9.1.2.3	Pad Driver Control	9-2
9.1.2.4	Port Control Coding	9-4
9.1.3	Port Register Description	9-5
9.1.3.1	Port Output Register	9-5
9.1.3.2	Port Output Modification Register	9-6
9.1.3.3	Port Input Register	9-7
9.1.3.4	Port Input/Output Control Registers	9-8
9.1.3.5	Pad Driver Mode Register	9-12
9.2	Pin Description	9-13
9.2.1	Device Pinning Overview	9-13
9.2.2	Port 0 Registers	9-13
9.2.2.1	Overview	9-13
9.2.2.2	Port 0 Register Implementations	9-15
9.2.2.3	Port 0 Functions	9-26
9.2.3	Port 1 Registers	9-29
9.2.3.1	Overview	9-29
9.2.3.2	Port 1 Register Implementations	9-30
9.2.3.3	Port 1 Functions	9-41
9.2.4	Port 2 Registers	9-45
9.2.4.1	Overview	9-45
9.2.4.2	Port 2 Register Implementations	9-46
9.2.4.3	Port 2 Functions	9-53
9.3	Address Map	9-56
10	Initialization of the SAK-CIC310-OSMX2HT Chip	10-1
10.1	Initialization for MLI Communication	10-2
10.1.1	The Initialization Values of Port 0	10-2
10.1.2	The Initialization Values of Port 1	10-2
10.1.3	The Initialization Values of Port 2	10-3
10.1.4	Initialization Values of MLI	10-3
10.1.5	The Initialization Values of the Service Request Control	10-3
10.2	Initialization for XMU Communication	10-4
10.2.1	The Initialization Values of Port 0	10-4
10.2.2	The Initialization Values of Port 1	10-4
10.2.2.1	XMU with 16-bit Data Width	10-5
10.2.3	The Initialization Values of Port 2	10-5

10.2.4	Initialization Values of the XMU	10-6
10.2.5	The Initialization Values of the Service Request Control	10-6
10.3	Initialization for SSC/SPI Communication	10-6
10.3.1	The Initialization Values of Port 0	10-7
10.3.2	The Initialization Values of Port 1	10-7
10.3.2.1	SSC Full-Duplex Mode	10-7
10.3.2.2	SSC Half-Duplex Mode	10-8
10.3.3	The Initialization Values of Port 2	10-8
10.3.4	The Initialization Values of the SSC	10-9
10.3.5	The Initialization Values of the Service Request Control	10-10
11	Register Description	11-1
11.1	Overview	11-1
11.1.1	FlexRay Communication Controller (E-Ray) Registers	11-2
11.1.2	SCU Kernel Registers	11-14
11.1.3	DMA Registers	11-18
11.1.4	MLI Kernel Registers	11-25
11.1.5	SSC Kernel Registers	11-28
11.1.6	XMU Kernel Registers	11-29
11.1.7	Ports Registers	11-29
11.1.8	Reserved Address Ranges not Associated to Peripheral	11-31

1 Architecture Overview

This device is a standalone communication controller for the Infineon AUDIO-NG 32-bit microcontroller family devices, the Infineon (X)C166 16-bit microcontroller family devices, and other microcontroller. The major task is to provide powerful FlexRay v2.1 communication controller to these microcontroller families.

1.1 About this Document

This document is designed to be read primarily by design engineers and software engineers who need a detailed description of the interactions of the SAK-CIC310-OSMX2HT functional units, registers, instructions, and exceptions.

This SAK-CIC310-OSMX2HT User's Manual describes the features of the SAK-CIC310-OSMX2HT with respect to the TriCore Architecture. Where the SAK-CIC310-OSMX2HT directly implements TriCore architectural functions, this manual simply refers to those functions as features of the SAK-CIC310-OSMX2HT. In all cases where this User's Manual describes a SAK-CIC310-OSMX2HT feature without referring to the TriCore Architecture, this implies that the SAK-CIC310-OSMX2HT is a direct implementation of the TriCore Architecture.

Where the SAK-CIC310-OSMX2HT implements a subset of TriCore architectural features, this manual describes the SAK-CIC310-OSMX2HT implementation, and then describes how it differs from the TriCore Architecture. Differences between the SAK-CIC310-OSMX2HT and the TriCore Architecture are documented in the text covering each such subject.

1.1.1 Related Documentations

A complete description of the TriCore architecture is found in the document entitled "TriCore Architecture Manual".

1.1.2 Text Conventions

This document uses the following text conventions for named components of the SAK-CIC310-OSMX2HT:

- Functional units of the SAK-CIC310-OSMX2HT are given in plain UPPER CASE. For example: "The XMU enables external bus masters to access all internal on-chip devices connected to the Crossbar switch."
- Pins using negative logic are indicated by an overline. For example: "The single system reset function initializes the SAK-CIC310-OSMX2HT into a defined default state and is invoked by an external $\overline{\text{PORST}}$ reset (Power-on reset) indicated by hardware reset input after power-on".
- Bit fields and bits in registers are in general referenced as "Register name.Bit field" or "Register name.Bit". For example: "Bit STAT.BSY is not set until the first clock

Architecture Overview

edge at SCLK appears". Most of the register names contain a module name prefix, separated by a underscore character "_" from the real register name (for example, "SSC_CON", where "SSC" is the module name prefix, and "CON" is the kernel register name). In chapters describing the kernels of the peripheral modules, the registers are mainly referenced with their kernel register names. The peripheral module implementation sections mainly refer to the real register names with module prefixes.

- Variables used to describe sets of processing units or registers appear in mixed upper-and-lower-case font. For example, register name "CHCR0n" refers to multiple "CHCR0" registers with variable n. The bounds of the variables are always given where the register expression is first used (for example, "(n = 0-7)"), and are repeated as needed in the rest of the text.
- The default radix is decimal. Hexadecimal constants are suffixed with a subscript letter "H", as in 100_H. Binary constants are suffixed with a subscript letter "B", as in: 111_B.
- When the extent of register fields, groups of signals, or groups of pins are collectively named in the body of the document, they are given as "NAME[A:B]", which defines a range for the named group from B to A. Individual bits, signals, or pins are given as "NAME[C]" where the range of the variable C is given in the text. For example: OSCGAIN[1:0].
- Units are abbreviated as follows:
 - **MHz** = Megahertz
 - **μs** = Microseconds
 - **kbaud, kbit** = 1000 characters/bit per second
 - **Mbaud, Mbit** = 1,000,000 characters/bit per second
 - **Kbyte** = 1024 byte of memory
 - **Mbyte** = 1048576 byte of memory

In general, the k prefix scales a unit by 1000 whereas the K prefix scales a unit by 1024. Hence, the Kbyte unit scales the expression preceding it by 1024. The kbaud unit scales the expression preceding it by 1000. The M prefix scales by 1,000,000 or 1048576, and μ scales by 0.000001. For example, 1 Kbyte is 1024 byte, 1 Mbyte is 1024 x 1024 byte, 1 kbaud/kbit are 1000 characters/bit per second, 1 Mbaud/Mbit are 1000000 characters/bit per second, and 1 MHz is 1,000,000 Hz.
- Data format quantities are defined as follows:
 - **Byte** = 8-bit quantity
 - **Half-word** = 16-bit quantity
 - **Word** = 32-bit quantity
 - **Double-word** = 64-bit quantity
- The SAK-CIC310-OSMX2HT supports only little-endian byte ordering (the least significant bytes are at lower addresses) for data, memory and registers.

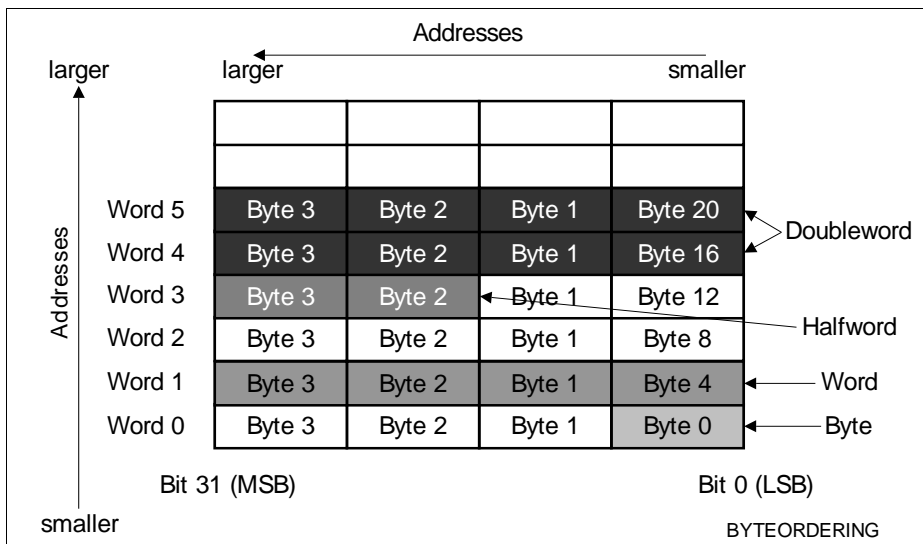


Figure 1-1 Byte Ordering

1.1.3 Reserved, Undefined, and Unimplemented Terminology

In tables where register bit fields are defined, the following conventions are used to indicate undefined and unimplemented functions. Furthermore, types of bits and bit fields are defined using the abbreviations as shown in [Table 1-1](#).

Table 1-1 Bit Function Terminology

Function of Bits	Description
Unimplemented, Reserved	<p>Register bit fields named 0 indicate unimplemented functions with the following behavior.</p> <ul style="list-style-type: none"> - Reading these bit fields returns 0. - Writing these bit fields has no effect. <p>These bit fields are reserved. When writing, software should always set such bit fields to 0 in order to preserve compatibility with future products.</p>
Undefined, Reserved	<p>Certain bit combinations in a bit field can be marked “Reserved”, indicating that the behavior of the SAK-CIC310-OSMX2HT is undefined for that combination of bits. Setting the register to such undefined bit or bit field combinations may lead to unpredictable results. Such bit combinations are reserved. When writing, software must always set such bit fields to legal values as defined for it.</p>
rw	The bit or bit field can be read and written.
rwh	As rw, but bit or bit field can be also set or reset by hardware.
r	The bit or bit field can only be read (read-only).
w	The bit or bit field can only be written (write-only).
rh	The bit or bit field can only be read. It can be also set or reset by hardware (typical example: status flags).
s	<p>Bits with this attribute are “sticky” in one direction. If there reset value is once overwritten by software, they can be switched again into there reset state only by a reset operation. Software cannot switch this type of bit into its reset state by writing the register. This attribute can be combined to “rws” or “rwhs”.</p>
f	<p>Bits with this attribute are readable only when they are accessed by an instruction fetch. Normal data read operations will return other values.</p>

1.1.4 Register Access Modes

Read and write access to registers and memory locations are sometimes restricted. In memory and register access tables, the terms as defined in [Table 1-2](#) are used.

Table 1-2 Access Terms

Symbol	Description
U	Access Mode: Access permitted in User Mode (not supported for SAK-CIC310-OSMX2HT). Reset Value: Value or bit is not changed by a reset operation.
SV	Access permitted in Supervisor Mode (only mode of SAK-CIC310-OSMX2HT).
R	Read-only register.
32	Only 32-bit word accesses are permitted to this register/address range.
E	Endinit protected register/address.
NC	No change, indicated register is not changed.
BE	Indicates that an access to this address range generates a Bus Error.
nBE	Indicates that no Bus Error is generated when accessing this address range, even though it is either an access to an undefined address or the access does not follow the given rules.
X	Undefined value or bit.

Note: All accesses of the SAK-CIC310-OSMX2HT, regardless of the host interface in use, are executed in Supervisor Mode.

1.1.5 Abbreviations

The following acronyms and termini are used in this document:

BCU	Bus Control Unit
CPU	Central Processing Unit (Host Processor)
DMA	Direct Memory Access
EBU	External Bus Unit
ERAY	FlexRay Protocol Controller
FPI	Flexible Peripheral Interconnect (Bus)
GPR	General Purpose Register
I/O	Input / Output
MLI	Micro Link Interface
NMI	Non-Maskable Interrupt
PLL	Phase Locked Loop
RAM	Random Access Memory
SCU	System Control Unit
SFR	Special Function Register
SRAM	Static Data Memory
SSC	Synchronous Serial Controller
XMU	External Memory Interface
ASC	Asynchronous/Synchronous Serial Interface
AP	Action Point
BD	Bus Driver
CAS	Collision Avoidance Symbol
CC	Communication Controller
CHI	Controller Host Interface
CIF	Customer Interface
CRC	Cyclic Redundancy Check
DTS	Dynamic Trailing Sequence
ECU	Electronic Control Unit
EMC	Electro-Magnetic Compatibility
FES	Frame End Sequence

FIFO	First In First Out (message buffer structure)
FSM	Finite State Machine
FSP	Frame and Symbol Processing
FSS	Frame Start Sequence
FTDMA	Flexible Time Division Multiple Access (media access method)
FTM	Fault Tolerant Midpoint
GIF	Generic Interface
GTU	Global Time Unit
IBF	Input Buffer
IFG	Inter Frame Gap
ISG	Inter Slot Gap
INT	Interrupt Control
MHD	Message Handler
MT	Macro tick
MTS	Media Access Test Symbol
NCT	Network Communication Time
NEM	Network Management
NIT	Network Idle Time
NM	Network Management
NRZ	Non-Return to Zero (method of encoding)
OBF	Output Buffer
POC	Protocol Operation Control
PRT	Protocol Controller
SDL	Specification and Description Language
SUC	System Universal Control
SW	Symbol Window
T.B.D.	To Be Defined
TBF	Transient Buffer
TDMA	Time Division Multiple Access (media access method)
TRP	Time Reference Point
TSS	Transmission Start Sequence
TT-D	Time Triggered Distributed Synchronization (protocol mode)

μ T	Microtick
VCW	Validation Check Window
WU	Walk-up
WUP	Walk-up Pattern
WUS	Walk-up Symbol

1.2 Summary of Features

The major functions supported by the SAK-CIC310-OSMX2HT are summarized in this section.

- A powerful FlexRay v2.1 Protocol Controller
 - Certified to be conform with FlexRay protocol specification v2.1
 - Data rates of up to 10 MBit/s on each channel
 - Up to 128 message buffers configurable
 - 8 Kbyte of Message RAM for storage of e.g. 128 message buffers with maximum 48 Byte data field or up to 30 messages with 254 Byte data field
 - Configuration of message buffers with different payload lengths
 - One configurable receive FIFO
 - Each message buffer can be configured as Receive Buffer, as Transmit Buffer, or as part of the receive FIFO
 - Host access to message buffers via Input and Output Buffer.
Input Buffer: holds message to be transferred to the Message RAM
Output Buffer: holds message read from the Message RAM
 - Filtering for frame ID, channel ID, and cycle counter
 - Network Management supported
 - Two Channels enabling one redundant FlexRay Bus
 - All data formats are little-endian
- Independant 8-Channel DMA Controller
 - 8 selectable request inputs per DMA channel
 - Programmable priority of DMA channels within the DMA sub-block (2 levels)
 - Software and hardware DMA request generation
 - Hardware requests by selected peripherals
 - Individually programmable operation modes for each DMA channel
 - Single Mode: stops and disables DMA channel after a predefined number of DMA transfers
 - Continuous Mode: DMA channel remains enabled after a predefined number of DMA transfers; DMA transaction can be repeated.
 - Programmable address modification
 - Support of circular buffer addressing mode
 - Programmable data width of a DMA transaction: 8-bit, 16-bit, or 32-bit
 - Individual register set for each DMA channel
 - Source and destination address register
 - Channel control and status register
 - Transfer count register
 - Flexible interrupt generation
- 16-bit External Memory Interface Unit (XMU)
 - 16-bitwide data bus (D[15:0])
 - Automatic data assembly/disassembly operation
 - Data width of external bus master can be 8-bit or 16-bit

Architecture Overview

- Automatic data assembly/disassembly operation
- 13-bitwide address bus (A[12:0])
- Address extension mechanism to 32-bit
- Read (RD) and write (WR) Bus control signal
- External synchronous/asynchronous wait state bus control signal (WAIT)
- External master chip select (CSFPI) to access on-chip devices connected to the crossbar switch
- High performing on-chip crossbar bus structure
 - 32-bit crossbar slave interface for FlexRay
 - 32-bit crossbar slave interface for Ports and System Control
 - 32-bit crossbar slave interface for MLI communication
 - 32-bit crossbar slave interface for MLI and DMA peripheral
 - 32-bit crossbar master interface for Host Communication Interfaces
 - 32-bit crossbar master interface for DMA
- Versatile High-Speed Synchronous Serial Channels (SSC) for Host Communication
 - Full-duplex or half-duplex operation
 - Automatic half-duplex pad control
 - SSC supports proprietary protocol to drive an integrated move engine
 - Baud rate: $f_{SSC} / 2$
Maximum baud rate of 40 MBit/s (@ 80 MHz module clock)
- Versatile High-Speed Micro Link interfaces (MLI) for serial inter-processor communication and Host Communication
 - Fully transparent read/write access supported (including remote programming)
 - Complete address range of target controller available
 - Special protocol to transfer data, address offset, or address offset and data
 - Error control using a parity bit
 - 32-bit, 16-bit, and 8-bit data transfers
 - Address offset width: from 1 to 16 bit
 - Baud rate: $f_{MLI} / 2$ (symmetric shift clock approach),
baud rate definition by the corresponding fractional divider
Maximum baud rate of 40 MBit/s (@ 80 MHz module clock)
- Full automotive temperature range: -40° to +125°C
- 26 digital general purpose I/O lines, 20 digital general purpose input lines
- Digital I/O ports with 3.3 V capability
- Clock Generation Unit with PLL
- Core supply voltage of 1.5 V
- I/O voltage of 3.3 V
- One Package Option only (P-TQFP-64)

As the FlexRay Protocol Controller is on a separate chip, the so called standalone Communication Controller, the access is handled via serial or parallel communication links. These three types of link options are discussed in the following chapter.

1.3 System Integration Concept

The SAK-CIC310-OSMX2HT IC is supposed to be connected to devices of the Infineon AUDO-NG 32bit microcontroller device family. The connection has to be done in a way that today's AUDO-NG 32-bit microcontroller MLI, ASC, or SSC interfaces remain untouched. On the other hand the FlexRay Communication controller requires communication bandwidth of 10 MBit/s and more to download the full message bandwidth of the two FlexRay links. Beside this requirement additionally the SAK-CIC310-OSMX2HT needs to be operative with a minimum pin account and area. Therefore two small serial interfaces (MLI, SSC) and an additional parallel interface are implemented.

For the SAK-CIC310-OSMX2HT concept three options are implemented in parallel to minimize the risk and to increase the options for a system composed out of the SAK-CIC310-OSMX2HT and an AUDO-NG 32-bit microcontroller device and still to handle the application requirements. These three interfaces are the SSC Interface (usable as SPI and synchronous ASC Interface), a parallel Interface, and the Micro Link Interface (MLI). All AUDO-NG 32-bit microcontroller family member are equipped with serial interfaces (ASC and SSC).

Three options are presented for the system solution based on these three communication interfaces.

1.3.1 MLI Host Link (Option One)

The first and preferred option is to connect the SAK-CIC310-OSMX2HT via the MLI interface to the AUDO-NG 32bit microcontroller family. The MLI has the advantage realizing a fast and smart communication interface with a low pin count and SPI like SW protocol handling. The transmission rate is high enough to handle the data traffic that is generated from the SAK-CIC310-OSMX2HT FlexRay Protocol engine to the AUDO-NG 32-bit microcontroller device and vice versa.

MLI Feature Set

- Serial communication from the MLI transmitter to MLI receiver of another controller
- Fully transparent read/write access supported (including remote programming)
- Complete address range of target controller available
- Special protocol to transfer data, address offset, or address offset and data
- Error control using a parity bit
- 32-bit, 16-bit, and 8-bit data transfers
- Address offset width: from 1 to 16 bit
- Baud rate: $f_{\text{MLI}} / 2$ (symmetric shift clock approach),
baud rate definition by the corresponding fractional divider
Maximum baud rate of 40 MBit/s (@ 80 MHz module clock)

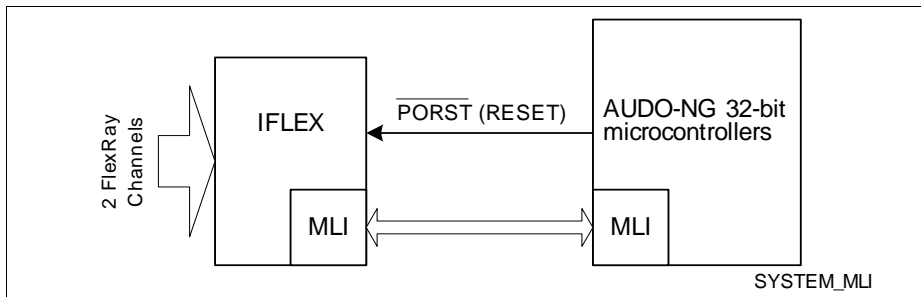


Figure 1-2 SAK-CIC310-OSMX2HT and AUDIO-NG 32bit Microcontroller Device Connected via MLI

The connectivity features of the MLI enables also connecting two different SAK-CIC310-OSMX2HT devices with the host. For further information see [Figure 7-10](#) This requires two sets of MLI interface signals bonded out of the host MLI.

1.3.2 SSC Host Link (Option Two)

The second option is to connect the SAK-CIC310-OSMX2HT via a SSC (SPI) interface to e.g. the AUDIO-NG 32-bit microcontroller family or the (X)C166 16-bit microcontroller family. Pin count are in a similar range compared with the MLI interface. The SW handling cost more host performance than the MLI interface requires. The single SPI link does not provide enough bandwidth to handle the maximum bandwidth FlexRay may require, but fulfil the needs of applications with lower bandwidth requirements.

SSC Feature Set

- Master and slave mode operation
 - Full-duplex or half-duplex operation
 - Automatic pad control possible
- Flexible data format
 - Programmable number of data bits: 2 to 16-bit
 - Programmable shift direction: LSB or MSB shift first
 - Programmable clock polarity: idle low or high state for the shift clock
 - Programmable clock/data phase: data shift with leading or trailing edge of the shift clock
- Baud rate: $f_{SSC} / 2$;
Maximum slave mode baud rate of 20 MBit/s (@ 80 MHz module clock)
- Interrupt generation
 - On a transmitter empty condition
 - On a receiver full condition
 - On an error condition (receive, phase, baud rate, transmit error)

Architecture Overview

- SSC supports proprietary protocol to drive an integrated move engine.

Note: Even so the SSC can be flexible configured, the SSC move engine can only handle 16 bit, Slave Mode, Full-duplex mode and Half-duplex mode, leading edge is high to low, shift on leading edge.

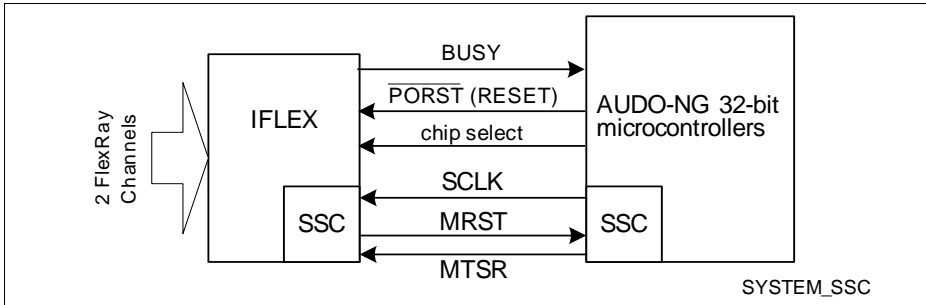


Figure 1-3 SAK-CIC310-OSMX2HT and AUDIO-NG 32bit Microcontroller Device Connected by SPI

The SSC is configured in a manner it could communicate to the ASC of the TriCore family. The TriCore ASC is configured in a manner using both RXDxA and RXDxB of the ASC as MRST and MTSR and the TXDxA or TXDxB as Clock.

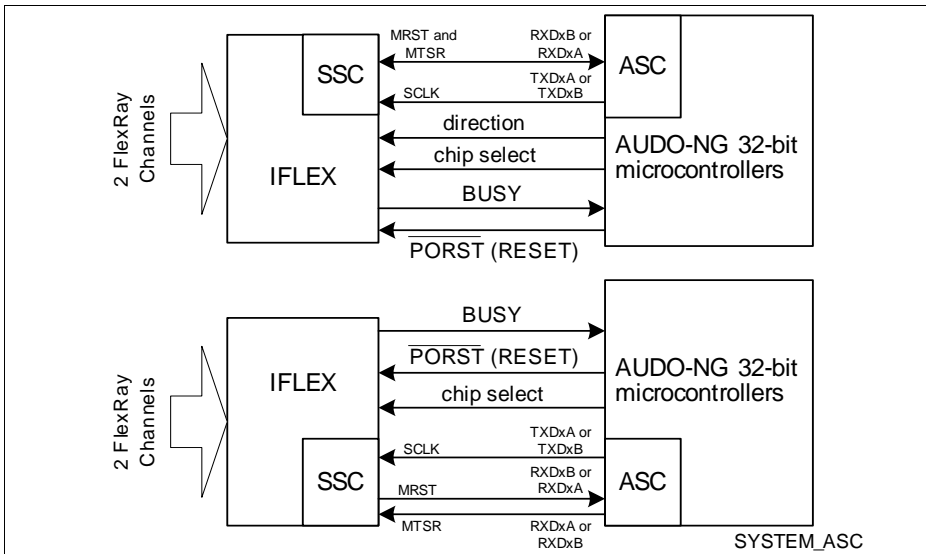


Figure 1-4 SAK-CIC310-OSMX2HT and AUDIO-NG 32bit Microcontroller Device Connected via ASC

1.3.3 Parallel Host Link (Option Three)

The third option is to connect the SAK-CIC310-OSMX2HT via the parallel interface e.g. the AUDO-NG 32-bit microcontroller family or the (X)C166 16-bit microcontroller family. The parallel interface provides enough bandwidth to handle the maximum data traffic generated by the SAK-CIC310-OSMX2HT FlexRay Protocol Controller to the AUDO-NG 32-bit microcontroller device and vice versa.

XMU Features

- 16-bit wide data bus (D[15:0])
 - Data width of external bus master can be 8 or 16 bit.
 - Automatic data assembly/disassembly operation
- 13-bit wide address bus (A[12:0])
- Bus control signals
 - Read (\overline{RD}) and write (\overline{WR})
 - Two byte control signals ($\overline{BC}[1:0]$)
 - External synchronous/asynchronous wait state control (\overline{WAIT})
 - External master chip select (\overline{CSFPI}) to access on-chip devices connected to the MIF Bus

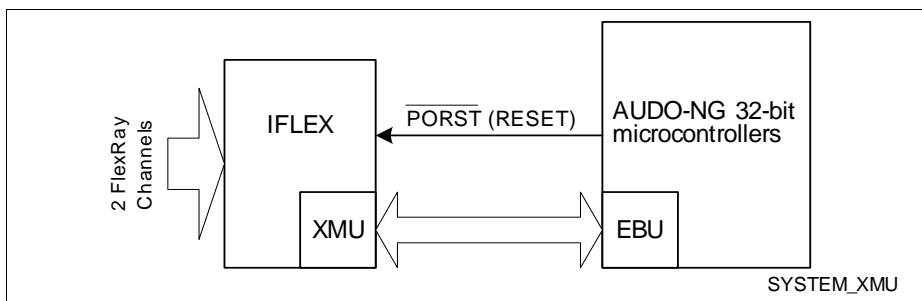


Figure 1-5 SAK-CIC310-OSMX2HT and AUDIO-NG 32bit Microcontroller Device Connected via XMU

1.3.4 DMA Controller

The Direct Memory Access (DMA) Controller of the SAK-CIC310-OSMX2HT transfers data from data source locations to data destination locations without intervention of the Host Controller. One data move operation is controlled by one DMA channel. Eight DMA channels are provided. The Bus Switch provides the connection of the DMA Sub-Block to the ERAY Peripheral, Host Communication peripheral, Ports, and an MLI bus interface. **Figure 1-6** shows the implementation details and interconnections of the DMA module.

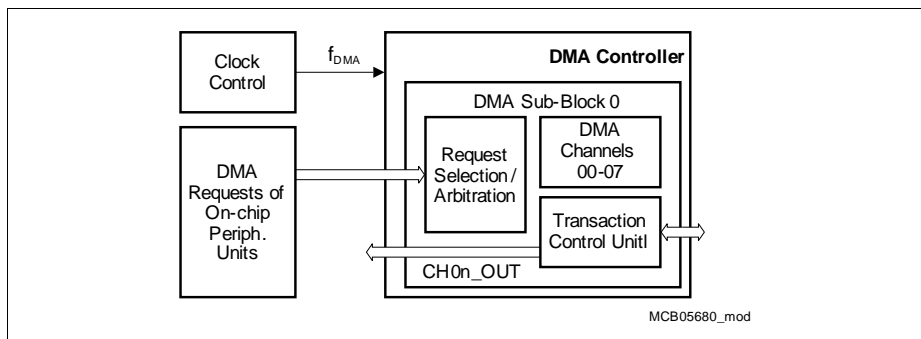


Figure 1-6 DMA Controller Block Diagram

Features

- 8 independent DMA channels
 - Up to 8 selectable request inputs per DMA channel
 - 2-level programmable priority of DMA channels within a DMA Sub-Block
 - Software and hardware DMA request
 - Hardware requests by selected on-chip peripherals and external inputs
- Programmable priority of the DMA Sub-Blocks on the crossbar switch
- Buffer capability for move actions on the buses (at least 1 move per bus is buffered).
- Individually programmable operation modes for each DMA channel
 - Single mode: stops and disables DMA channel after a predefined number of DMA transfers
 - Continuous mode: DMA channel remains enabled after a predefined number of DMA transfers; DMA transaction can be repeated.
 - Programmable address modification
- Full 32-bit addressing capability of each DMA channel
 - 4 GByte address range
 - Support of circular buffer addressing mode
- Programmable data width of DMA transfer/transaction: 8-bit, 16-bit, or 32-bit
- Micro Link bus interface support
- Register set for each DMA channel
 - Source and destination address register
 - Channel control and status register
 - Transfer count register
- Flexible interrupt generation (the service request node logic for the MLI channels is also implemented in the DMA module)
- Read/write requests of the System Bus Side to the Remote Peripherals are bridged to the Remote Peripheral Bus (only the DMA is master on the RPB)

1.4 General Device Information

This section provides an overview of the entire architecture of the SAK-CIC310-OSMX2HT companion chip.

The overall building blocks of the SAK-CIC310-OSMX2HT are:

- FlexRay v2.1 protocol controller
- Slave (SPI) SSC interface
- MLI interface
- Parallel external memory interface unit (XMU).
- 8 independant channel DMA
- Dual Voltage Power Supply
- One Package Option only (P-TQFP-64)

1.4.1 Block Diagram

Figure 1-7 summarizes the overall architecture of the SAK-CIC310-OSMX2HT.

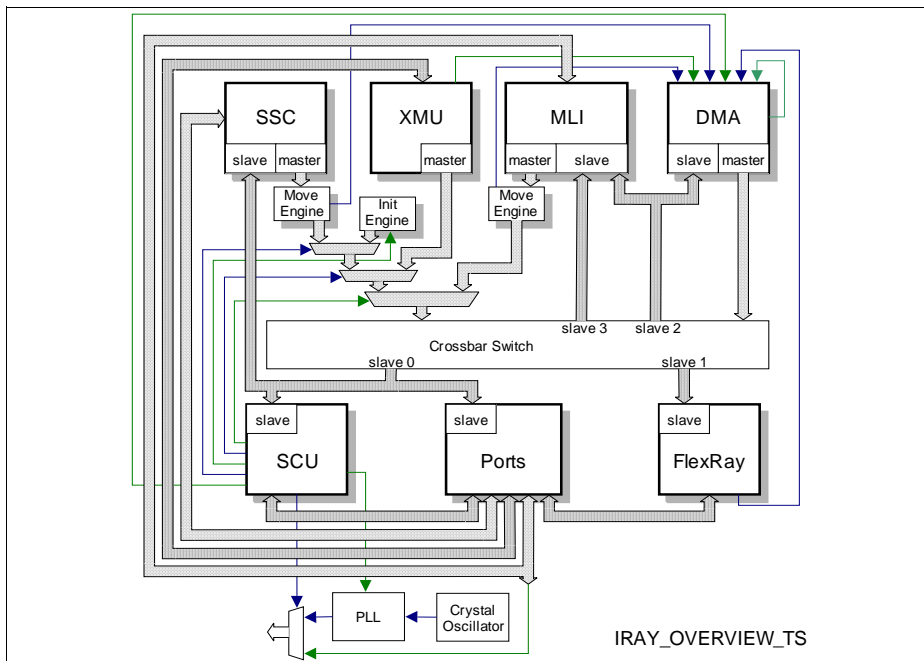


Figure 1-7 IFLEX Block Diagram

In **Figure 1-7** the block diagram of the IFLEX is shown. This concept allows the access to the FlexRay Protocol Controller for the host CPU without sacrificing any of the features

Architecture Overview

of the FlexRay Protocol Controller. This can be achieved, as all registers of the FlexRay Protocol Controller are mapped to the crossbar switch. This crossbar switch can be accessed via one of the three host interfaces, which were introduced in [Chapter 1.3](#). The interface selection is done via the mode signals MODE[0] and MODE[1], which can be directly connected to the supply voltage or via pull-up/down resistors (of about 10-47 kΩ). The cross bar switch allows two parallel data operations, one initiated by the DMA, the other by either the MLI, XMU, or SSC move engine and serving different ports of the crossbar switch to slave interfaces. The address ranges of the slave ports are:

1. Slave Port 0 serves the address range from 0000 0800_H to 0000 0FFF_H
2. Slave Port 1 serves the address range from 0000 1000_H to 0000 1FFF_H
3. Slave Port 2 serves the address range from 0000 0200_H to 0000 07FF_H
4. Slave Port 3 serves all addresses 0000 0000_H to 0000 01FF_H and 0000 2000_H to FFFF FFFF_H

So apart from a possible loss of speed due to the serial interfaces the complete functionality of the FlexRay Protocol Controller is maintained.

The crossbar switch domain is completely separated from the address domain on the CPU chip. The addresses of all modules on the FlexRay Communication Controller are 32-bit addresses. Transactions between the CPU and the SSC are executed with the SSC transmission protocol and a high level protocol to drive the move engine, transactions between the MLI and the CPU use the MLI transmission protocol and transactions between the XMU and the CPU are based on the XMU transmission protocol.

Each transaction via any of the three host interfaces is defined by address, data, data width and type of frame. The address, where data is read from or written to, is related to the crossbar switch address domain. The data width may be 8, 16, or 32 bit for the MLI, 8 or 16 bit for the XMU, and 16 bit for the SSC. The type of frame may be a read or write access or an answer frame to a read access. The Move Engines request access to the bus via the Bus Master. The Flex Protocol Controller and the MLI may send an interrupt to the DMA.

There are several cases:

- The MLI, XMU, or the SSC interfaces requests to write.
- The MLI, XMU, or the SSC interfaces requests to read.
- The FlexRay Communication Controller requests to write.
- A general peripheral beside the FlexRay Communication Controller requests to read.

Half-word (16-bit) accesses, read and write, are only allowed for half-word aligned addresses (even addresses). Therefore if doing a half-word (16-bit) access, the least significant address bit (A0) is ignored (assumed to be 0) by the modules of the SAK-CIC310-OSMX2HT. Therefore if addressing a half-word with an odd address n, a half-word is read instead from address n-1 (ignoring least significant address bit: n AND FFFF FFFE_H).

Architecture Overview

Word accesses (32-bit), read and write, are only allowed for word aligned addresses (addresses modulo 4 = 0). Therefore if doing a word (32-bit) access, the two least significant address bits (A0 and A1) are ignored (assumed to be to 0) by the modules of the SAK-CIC310-OSMX2HT. Therefore if addressing a word with an address n ($n \bmod 4 = \{1,2,3\}$), a word is read instead from address ($n \text{ AND } \text{FFFF FFFC}_H$) (ignoring the two least significant address bits).

1.4.2 Pin Definition and Functions

To enable a clean power-up, the majority of pins have a “enable pad supply” (ENPS) function. This ENPS is pad supply (V_{DDP}) driven. So only if the pad supply is stable, the output pads are activated. Five pins do not support ENPS: JTAGEN, PORST, XTAL1, and XTAL2. In case of a low voltage signal at the input port ENPS the output port PAD activates the weak pull-up and disables the output driver independent of the port direction. In case of a high voltage signal at the input port ENPS the bidirectional platform pads operate in normal mode.

1.4.2.1 Package Outline

The package outline and the signals of the pins are summarized in [Figure 1-8](#).

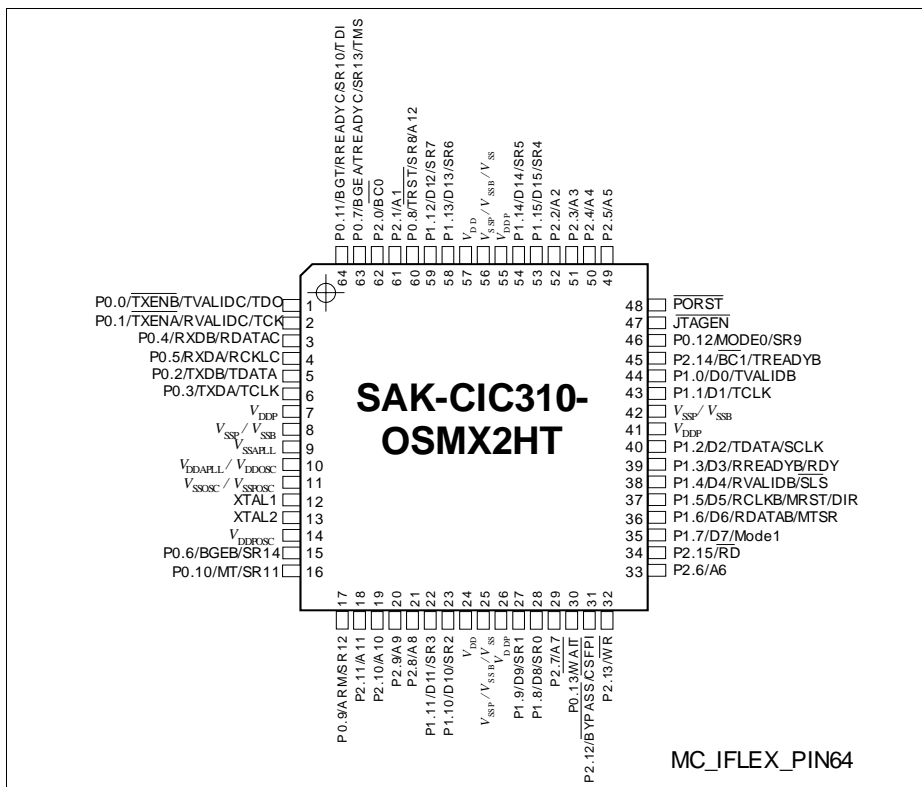


Figure 1-8 SAK-CIC310-OSMX2HT Pinning: P-TQFP-64 Package (top view)

1.4.2.2 Pin Description

Table 1-3 Pin Definitions and Functions

Symbol	Pin	I/O	Function
	P-TQFP 64		
FlexRay Bus Interface			
TXENA/ TCK	2	I/O	Port 0 line 1 FlexRay Transmit Enable (Channel A) JTAG Module Clock Input (TCK) JTAG Enable Mode only
TXENB/ TDO	1	I/O	Port 0 line 0 FlexRay Transmit Enable (Channel B) JTAG Module Serial Data Output JTAG Enable Mode only
RXDA	4	I	Port 0 line 5 FlexRay Data Receiver Input (Channel A)
RXDB	3	I	Port 0 line 4 FlexRay Data Receiver Input (Channel B)
TXDA	6	I/O	Port 0 line 3 FlexRay Data Transmitter Output (Channel A)
TXDB	5	I/O	Port 0 line 2 FlexRay Data Transmitter Output (Channel B)
BGEA/ TMS/ SR13	63	I	Port 0 line 7 FlexRay Bus Guardian Enable (Channel A) JTAG Module State Machine Control Input JTAG Enable Mode only Interrupt Request input line 13 The logic 0 level at this pin indicates an interrupt request from the external host device.

Table 1-3 Pin Definitions and Functions (cont'd)

Symbol	Pin	I/O	Function
	P-TQFP 64		
BGEB/ SR14	15	I	Port 0 line 6 FlexRay Bus Guardian Enable (Channel B) Interrupt Request input line 14 The logic 0 level at this pin indicates an interrupt request from the external host device.
MT/ SR11	16	I/O	Port 0 line 10 FlexRay Bus Guardian Macro Tick FlexRay corrected Macro Tick Clock Interrupt Request Input/Output line 11 The logic 0 level at this pin indicates an interrupt request from/to the external host device.
BGT/ TDI/ SR10	64	I/O	Port 0 line 11 FlexRay Bus Guardian Tick Used by the Macro Tick Watchdog of the Bus Guardian as Time Base JTAG Module Serial Data Input JTAG Enable Mode only. Interrupt Request Input/Output line 10 The logic 0 level at this pin indicates an interrupt request from/to the external host device.
ARM/ SR12	17	I/O	Port 0 line 9 FlexRay Bus Guardian Arm Signal Indicates the begin of a communication cycle to the bus guardian. Interrupt Request Input/Output line 12 The logic 0 level at this pin indicates an interrupt request from/to the external host device.

Table 1-3 Pin Definitions and Functions (cont'd)

Symbol	Pin	I/O	Function
	P-TQFP 64		
Host Interfaces			
D0/ TVALIDB	44	I/O	Port 1 line 0 MODE = 00 _B : XMU Data bus Line 0 MODE = 01 _B : MLI Transmit Channel Valid Output MODE = 10 _B : XMU Data Bus line 0 MODE = 11 _B : Port 1 line 0 (Input/Output)
D1/ TCLK	43	I/O	Port 1 line 1 MODE = 00 _B : XMU Data Bus Line 1 MODE = 01 _B : MLI Transmit Channel Clock Output MODE = 10 _B : XMU Data Bus Line 1 MODE = 11 _B : Port 1 line 1 (Input/Output)
D2/ TDATA/ SCLK	40	I/O	Port 1 line 2 MODE = 00 _B : XMU Data Bus Line 2 MODE = 01 _B : MLI Transmit Channel Data Output MODE = 10 _B : XMU Data Bus Line 2 MODE = 11 _B : SSC Serial Channel Clock (Input/Output)

Table 1-3 Pin Definitions and Functions (cont'd)

Symbol	Pin	I/O	Function
	P-TQFP 64		
D3/ RREADYB/ RDY	39	I/O	Port 1 line 3 MODE = 00 _B : XMU Data Bus Line 3 MODE = 01 _B : MLI Receive Channel Ready Output B MODE = 10 _B : XMU Data Bus Line 3 MODE = 11 _B : SSC Ready Signal (Output) Output signal indicating that the standalone device is ready for data transfer.
D4/ RVALIDB/ SLS	38	I/O	Port 1 line 4 MODE = 00 _B : XMU Data Bus Line 4 MODE = 01 _B : MLI Receive Channel Valid Input B MODE = 10 _B : XMU Data Bus Line 4 MODE = 11 _B : SSC Select Slave Input used to enable SSC action when active.
D5/ RCLKB/ MRST/ DIR	37	I/O	Port 1 line 5 MODE = 00 _B : XMU Data Bus Line 5 MODE = 01 _B : MLI Receive Channel Clock Input B MODE = 10 _B : XMU Data Bus Line 5 MODE = 11 _B : SPI Master Receive Slave Transmit Serial data output Alternative: Direction of SPI Half-Duplex communication.

Table 1-3 Pin Definitions and Functions (cont'd)

Symbol	Pin	I/O	Function
	P-TQFP 64		
D6/ RDATA6/ MTSR	36	I/O	Port 1 line 6 MODE = 00 _B : XMU Data Bus Line 6 MODE = 01 _B : MLI Receive Channel Data Input B MODE = 10 _B : XMU Data Bus Line 6 MODE = 11 _B : SPI Master Transmit Slave Receive Serial data input
BC[1]/ TREADYB	45	I	Port 2 line 14 MODE = 00 _B : XMU Byte control line 1 Controls the byte access to corresponding byte location D[15:8] MODE = 01 _B : MLI Transmit Channel Ready Input B MODE = 10 _B : XMU Byte control line 1 Controls the byte access to corresponding byte location D[15:8] MODE = 11 _B : Port 2 line 14 (input only)
Parallel Host Interface			
BC[0]	62	I	Port 2 line 0 XMU Byte control line 0 Controls the byte access to corresponding byte location D[7:0]

Table 1-3 Pin Definitions and Functions (cont'd)

Symbol	Pin	I/O	Function
	P-TQFP 64		
A[11:1]		I	XMU
A1	61		Port 2 line 1 Address bus lines 1
A2	52		Port 2 line 2 Address bus lines 2
A3	51		Port 2 line 3 Address bus lines 3
A4	50		Port 2 line 4 Address bus lines 4
A5	49		Port 2 line 5 Address bus lines 5
A6	33		Port 2 line 6 Address bus lines 6
A7	29		Port 2 line 7 Address bus lines 7
A8	21		Port 2 line 8 Address bus lines 8
A9	20		Port 2 line 9 Address bus lines 9
A10	19		Port 2 line 10 Address bus lines 10
A11	18		Port 2 line 11 Address bus lines 11
D8/ SR0	28	I/O	XMU Port 1 line 8 Data bus lines 8 Interrupt Request Input/Output line 0 The logic 0 level at this pin indicates an interrupt request from/to the external host device.
D9/ SR1	27	I/O	XMU Port 1 line 9 Data bus lines 9 Interrupt Request Input/Output line 1 The logic 0 level at this pin indicates an interrupt request from/to the external host device.
D10/ SR2	23	I/O	XMU Port 1 line 10 Data bus lines 10 Interrupt Request Input/Output line 2 The logic 0 level at this pin indicates an interrupt request from/to the external host device.
D11/ SR3	22	I/O	XMU Port 1 line 11 Data bus lines 11 Interrupt Request Input/Output line 3 The logic 0 level at this pin indicates an interrupt request from/to the external host device.

Table 1-3 Pin Definitions and Functions (cont'd)

Symbol	Pin	I/O	Function
	P-TQFP 64		
D12/ SR7	59	I/O	XMU Port 1 line 12 Data bus lines 12 Interrupt Request Input/Output line 7 The logic 0 level at this pin indicates an interrupt request from/to the external host device.
D13/ SR6	58	I/O	XMU Port 1 line 13 Data bus lines 13 Interrupt Request Input/Output line 6 The logic 0 level at this pin indicates an interrupt request from/to the external host device.
D14/ SR5	54	I/O	XMU Port 1 line 14 Data bus lines 14 Interrupt Request Input/Output line 5 The logic 0 level at this pin indicates an interrupt request from/to the external host device.
D15/ SR4	53	I/O	XMU Port 1 line 15 Data bus lines 15 Interrupt Request Input/Output line 4 The logic 0 level at this pin indicates an interrupt request from/to the external host device.
RD	34	I	Port 2 line 15 XMU Read control line Active during read operation
WR	32	I	Port 2 line 13 XMU Write control line Active during write operation input line
WAIT	30	I/O	Port 0 line 13 XMU Wait output
Control Signals			
PORST	48	I	Power-on Reset

Table 1-3 Pin Definitions and Functions (cont'd)

Symbol	Pin	I/O	Function
	P-TQFP 64		
JTAGEN	47	I	JTAG Enabled Mode Selection/ Pin JTAGEN selects whether JTAG Enabled Mode is used to access the SAK-CIC310-OSMX2HT device. JTAGEN = 0 _B : JTAG Enabled Mode JTAGEN = 1 _B : Normal Mode
TRST/ SR8/ A12	60	I/O	Port 0 line 8 Test Reset (JTAG Enable Mode) Interrupt Request Input/Output Line 8 The logic 0 level at this pin indicates an interrupt request to/from the external host device. Alternative: Address bus lines 12
MODE[0]/ SR9 ¹⁾	46 ²⁾	I/O	Port 0 line 12 Interface Selection 0 Pin MODE 0 selects if the on-chip serial Host Communication Links (MLI or SSC) or the parallel XMU Host Communication Link is enabled for communicating to the SAK-CIC310-OSMX2HT device. MODE 0 = 0 _B : On-chip XMU MODE 0 = 1 _B : On-chip SSC or MLI After latching the initial state with the rising edge of the PORST signal (e.g. external pull-up or pull-down resistors), this MODE 0 pin can be used as: Interrupt Request Input/Output Line 9 The logic 0 level at this pin indicates an interrupt request to/from the external host device.

Table 1-3 Pin Definitions and Functions (cont'd)

Symbol	Pin	I/O	Function
	P-TQFP 64		
D7/ Mode[1]³⁾	35	I/O	<p>Port 1 line 7 MODE[0]= 0_B: XMU Data Bus Line 7 MODE[0]= 1_B: Interface Selection 1 Pin MODE 1 selects if the on-chip serial MLI Communication Link or the serial SPI Host Communication Link is enabled for communicating to the SAK-CIC310-OSMX2HT device if Mode[0] = 1. MODE[0]= 1 AND MODE[1] = 0_B: On-chip MLI MODE[0]= 1 AND MODE[1]= 1_B: On-chip SSC After latching the initial state with the rising edge of the $\overline{\text{PORST}}$ signal (e.g. external pull-up or pull-down resistors), this MODE 1 pin can be used as standard IO pin.</p>
<u>BYPASS</u> CSFP1⁴⁾	31	I	<p>Port 2 line 12 With the rising edge of the $\overline{\text{PORST}}$ the $\overline{\text{BYPASS}}$ signal is sampled: 0 The oscillator circuitry is bypassed and f_{osc} is directly derived from XTAL1 (OSCCON.OSCBY is set to 1). 1 In normal operating mode the oscillator is running and f_{osc} is derived from the crystal or from an external clock signal (OSCCON.OSCBY is set to 0). After latching the initial state with the rising edge of the $\overline{\text{PORST}}$ signal (e.g. external pull-up or pull-down resistors), this $\overline{\text{BYPASS}}$ pin can be used as standard Input pin as described following: MODE = 00_B: XMU Chip Select MODE = 01_B: Port 2 Line 12 (Input) MODE = 10_B: XMU Chip Select MODE = 11_B: Port 2 Line 12 (Input)</p>

Table 1-3 Pin Definitions and Functions (cont'd)

Symbol	Pin	I/O	Function
	P-TQFP 64		
V_{DDP}	7, 26, 41, 55	+3.3 V	Power Supply , supply for IO pads
$V_{SSP}^{5)}$	8,25,42 56	0V	Ground , for IO pads
$V_{SSB}^{5)}$	8,25,42 56	0V	Ground , for Bulk Contact
$V_{DDAPLL}^{5)}$	10	+1.5 V	Power Supply , supply for analogue PLL circuitries
V_{SSAPLL}	9	0V	Ground , for analogue PLL circuitries
$V_{DDOSC}^{5)}$	10	+1.5 V	Power Supply , supply for oscillator
$V_{SSOSC}^{5)}$	11	0V	Ground , for oscillator
$V_{SSOSC}^{5)}$	11	0V	Ground , for oscillator pad
V_{DDPOSC}	14	+3.3 V	Power Supply , supply for oscillator pad
V_{DD}	24, 57	+1.5 V	Power Supply , supply for digital module cores
$V_{SS}^{5)}$	25, 56	0V	Digital Ground , for digital module cores
XTAL1	12	I	XTAL1 Input of the inverting oscillator amplifier and input to the internal clock generation circuit. When the SAK-CIC310-OSMX2HT device is provided with an external clock, XTAL1 should be driven while XTAL2 is left unconnected. Minimum and maximum high and low pulse width as well as rise/fall times specified in the AC characteristics must be respected.
XTAL2	13	O	XTAL2 Output of the inverting oscillator amplifier.

1) The initial logic state on pins \overline{MODE} is latched while the \overline{PORST} input is active.

2) During Reset an internal pull-up device is connected.

3) The initial logic state on pins \overline{MODE} is latched while the \overline{PORST} input is active.

4) The initial logic state on pin \overline{BYPASS} is latched while the \overline{PORST} input is active.

5) Some power supplies are multiple bonded to a common pin.

In **Table 1-3** the signals and power lines going outside the standalone chip are summarized. These signals are partially routed via the pad logic to the pins. The analog signals go directly to the pins.

Figure 1-8 shows the pinning for a P-TQFP-64 pin package.

Architecture Overview

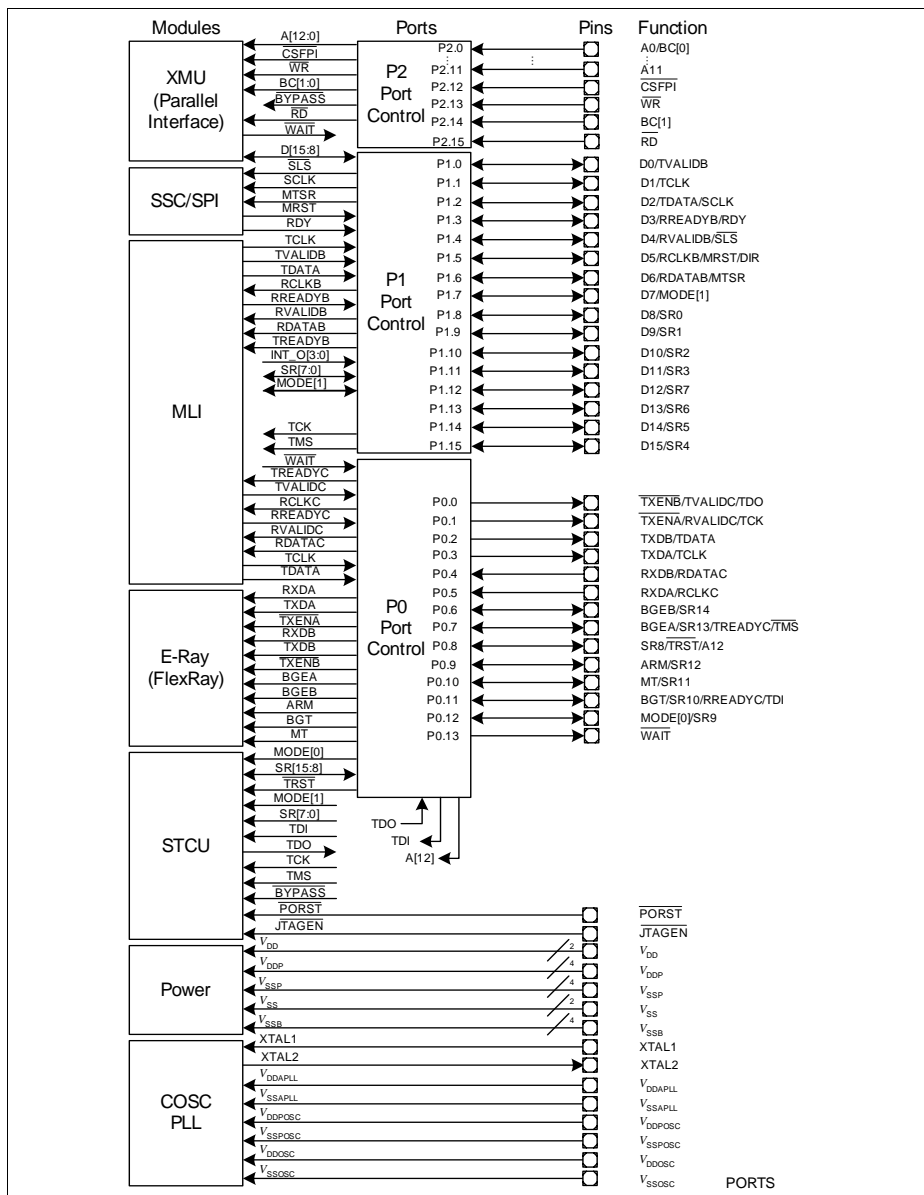


Figure 1-9 Pins for a P-TQFP- 64 Pin Package

1.4.2.3 Ordering Information

The ordering code for Infineon companion chip provides an exact reference to the required product. This ordering code identifies:

- The derivative itself, i.e. its function set, the temperature range, and the supply voltage
- The package and the type of delivery.

For the available ordering codes for the SAK-CIC310-OSMX2HT please refer to the **“Product Catalog Microcontrollers”**, which summarizes all available microcontroller variants.

This document describes the derivatives of the device. The [Table 1-4](#) enumerates these derivatives and summarizes the differences.

Table 1-4 SAK-CIC310-OSMX2HT Derivative Synopsis

Derivative	Ambient Temperature Range
SAK-CIC310-OSMX2HT	$T_A = -40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$

2 FlexRay Protocol Controller (E-Ray)

The E-Ray IP-module performs communication according to the FlexRay protocol specification v2.1. With maximum specified clock the bitrate can be programmed to values up to 10MBit/s. Additional bus driver (BD) hardware is required for connection to the physical layer.

2.1 Overview

For communication on a FlexRay network, individual message buffers with up to 254 data byte are configurable. The message storage consists of a single-ported Message RAM that holds up to 128 message buffers. All functions concerning the handling of messages are implemented in the Message Handler. Those functions are the acceptance filtering, the transfer of messages between the two FlexRay Channel Protocol Controllers and the Message RAM, maintaining the transmission schedule as well as providing message status information.

The register set of the E-Ray IP-module can be accessed directly by an external Host via the module's Host interface. These registers are used to control/configure/monitor the FlexRay Channel Protocol Controllers, Message Handler, Global Time Unit, System Universal Control, Frame and Symbol Processing, Network Management, Service Request Control, and to access the Message RAM via Input / Output Buffer.

The E-Ray IP-module can be connected to a wide range of customer-specific Hosts via its 8/16/32-bit Generic Host Interface.

The E-Ray IP-module supports the following features:

- Conformance with FlexRay protocol specification v2.1
- Data rates of up to 10Mbit/s on each channel
- Up to 128 message buffers configurable
- 8 Kbyte of Message RAM for storage of e.g. 128 message buffers with max. 48 byte data field or up to 30 message buffers with 254 byte data sections
- Configuration of message buffers with different payload lengths possible
- One configurable receive FIFO
- Each message buffer can be configured as receive buffer, as transmit buffer or as part of the receive FIFO
- Host access to message buffers via Input and Output Buffer.
Input Buffer: Holds message to be transferred to the Message RAM
Output Buffer: Holds message read from the Message RAM
- Filtering for slot counter, cycle counter, and channel
- Maskable module service requests
- Network Management supported
- Four service request lines
- Asynchronous reset

FlexRay Protocol Controller (E-Ray)

- Automatic delayed read access to Output Command Request Register (**OBCR**) if a data transfer from Message RAM to Output Shadow Buffer (initiated by a previous write access to the **OBCR**) is ongoing.
- Automatic delayed read access to Input Command Request Register (**IBCR**) if a data transfer from Input Shadow Buffer to Message RAM (initiated by a previous write access to the **IBCR**) is ongoing.

2.2 Definitions

FlexRay Frame: Header Segment + Payload Segment

Message Buffer: Header Section + Data Section

Message RAM: Header Partition + Data Partition

Data Frame: FlexRay frame that is not a null frame

2.3 Block Diagram

The ERAY is built up by the following main submodules:

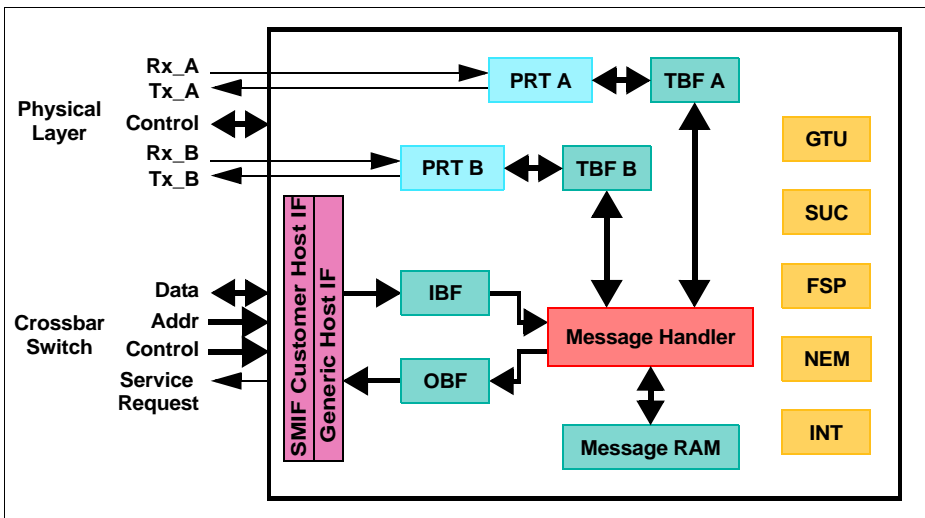


Figure 2-1 E-Ray Block Diagram

SMIF Customer Host Interface (SMIF CIF)

Connects the crossbar switch to the E-Ray IP-module via the Generic Host Interface.

FlexRay Protocol Controller (E-Ray)**Generic Host Interface (GIF)**

The E-Ray IP-module is provided with an 8/16/32-bit Generic Host Interface prepared for the connection to a wide range of customer-specific Hosts. Configuration registers, status registers, and service request registers are attached to the respective blocks and can be accessed via the Generic Host Interface.

Input Buffer (IBF)

For write access to the message buffers configured in the Message RAM, the Host can write the header and data section for a specific message buffer to the Input Buffer. The Message Handler then transfers the data from the Input Buffer to the selected message buffer in the Message RAM.

Output Buffer (OBF)

For read access to a message buffer configured in the Message RAM the Message Handler transfers the selected message buffer to the Output Buffer. After the transfer has completed, the Host can read the header and data section of the transferred message buffer from the Output Buffer.

Message Handler (MHD)

The E-Ray Message Handler controls data transfers between the following components:

- Input / Output Buffer and Message RAM
- Transient Buffer RAMs of the two FlexRay Protocol Controllers and Message RAM

Message RAM (MRAM)

The Message RAM consists of a single-ported RAM that stores up to 128 FlexRay message buffers together with the related configuration data (header and data partition).

Transient Buffer RAM (TBF 1/2)

Stores the data section of two complete messages.

FlexRay Channel Protocol Controller (PRT A/B)

The FlexRay Channel Protocol Controllers consist of shift register and FlexRay protocol FSM. They are connected to the Transient Buffer RAMs for intermediate message storage and to the physical layer via bus driver BD.

They perform the following functionality:

- Control and check of bit timing
- Reception and transmission of FlexRay frames and symbols
- Check of header CRC
- Generation / check of frame CRC

FlexRay Protocol Controller (E-Ray)

- Interfacing to bus driver

The FlexRay Channel Protocol Controllers have interfaces to:

- Physical Layer (bus driver)
- Transient Buffer RAM
- Message Handler
- Global Time Unit
- System Universal Control
- Frame and Symbol Processing
- Network Management
- Service Request Control

Global Time Unit (GTU)

The Global Time Unit performs the following functions:

- Generation of microtick
- Generation of macrotick
- Fault tolerant clock synchronization by FTM algorithm
 - Rate correction
 - Offset correction
- Cycle counter
- Timing control of static segment
- Timing control of dynamic segment (minislottling)
- Support of external clock correction

System Universal Control (SUC)

The System Universal Control controls the following functions:

- Configuration
- Wakeup
- Startup
- Normal Operation
- Passive Operation
- Monitor Mode

Frame and Symbol Processing (FSP)

The Frame and Symbol Processing controls the following functions:

- Checks the correct timing of frames and symbols
- Tests the syntactical and semantical correctness of received frames
- Sets the slot status flags

Network Management (NEM)

Handles of the network management vector

Service Request Control (INT)

The Service Request Controller performs the following functions:

- Provides error and status service request flags
- Enables and disables service request sources
- Assignment of service request sources to one of the two module service request lines
- Enables and disables module service request lines
- Manages the two service request timers
- Stop watch time capturing

2.4 Programmer's Model

The programmer's model of the E-Ray module follows the principle of memory mapped peripheral. Some portion of the memory follows the principle of segmented/paged memory organization.

2.4.1 Register Map

The E-Ray module allocates an address space of 2 Kbyte (000_H to 7FF_H). The registers are organized as 32-bit registers. 8/16-bit accesses are also supported. Host access to the Message RAM is done via the Input and Output Buffers. They buffer data to be transferred to and from the Message RAM under control of the Message Handler, avoiding conflicts between Host accesses and message reception / transmission. Addresses 0000_H to 000F_H are reserved for customer specific purposes. All functions related to these addresses are located in the Customer Host Interface. The test registers located on address 0010_H and 0014_H are writeable only under the conditions described in **“Special Registers” on Page 2-18**.

The assignment of the message buffers is done according to the scheme shown in **Table 2-1** below. The number N of available message buffers depends on the payload length of the configured message buffers. The maximum number of message buffers is 128. The maximum payload length supported is 254 byte.

The message buffers are separated into three consecutive groups:

- Static Buffers: Transmit / Receive Buffers assigned to static segment
- Static and Dynamic Buffers: Transmit / Receive Buffers assigned to static or dynamic segment
- FIFO- Receive FIFO

The message buffer separation configuration can be changed only in “DEFAULT_CONFIG” or “CONFIG” state only by programming the Message RAM Configuration register (**MRC**).

The first group starts with message buffer 0 and consists of static message buffers only. Message buffer 0 is dedicated to hold the startup / sync frame or the single slot frame, if node transmit one, as configured by **SUCC1.TXST**, **SUCC1.TXSY**, and **SUCC1.TSM** in

FlexRay Protocol Controller (E-Ray)

the SUC Configuration Register 1 (**SUCC1**). In addition, message buffer 1 may be used for sync frame transmission in case that sync frames or single-slot frames should have different payloads on the two channels. In this case bit **MRC.SPLM** has to be programmed to 1 and message buffers 0 and 1 have to be configured with the key slot ID and can be (re)configured in “DEFAULT_CONFIG” or “CONFIG” state only.

The second group consists of message buffers assigned to the static or to the dynamic segment. Message buffers belonging to this group may be reconfigured during run time from dynamic to static or vice versa depending on the state of **MRC.SEC**.

The message buffers belonging to the third group are concatenated to a single receive FIFO.

Table 2-1 Assignment of Message Buffers

Message Buffer 0	↓ Static Buffers	
Message Buffer 1		
...		
	↓ Static + Dynamic Buffers	← FDB
	↓ FIFO	← FFB
Message Buffer N-1		
Message Buffer N		← LCB

2.4.2 E-Ray Kernel Registers

This chapter describes all registers of the E-Ray kernel.

Table 2-2 Registers Address Space E-Ray Kernel Register Address Space

Module	Base Address	End Address	Note
ERAY	00001000 _H	000017FF _H	2Kbyte

Table 2-3 Registers OverviewE-Ray Kernel Registers

Register Short Name	Register Long Name	Address Offset	Details see
---------------------	--------------------	----------------	-------------

Customer Registers

CUST0	Product Identification Register	0000 0000 _H	Page 2-11
-------	---------------------------------	------------------------	---------------------------

FlexRay Protocol Controller (E-Ray)
Table 2-3 Registers OverviewE-Ray Kernel Registers (cont'd)

Register Short Name	Register Long Name	Address Offset	Details see
CUST1	Busy Control Register	0000 0004 _H	Page 2-13
ID	Module Identification Register	0000 0008 _H	Page 2-12
CUST3	Customer Interface Timeout Counter	0000 000C _H	Page 2-15

Special Registers

TEST1	Test Register 1	0000 0010 _H	Page 2-18
TEST2	Test Register 2	0000 0014 _H	Page 2-23
-	Reserved	0000 0018 _H	-
LCK	Lock Register	0000 001C _H	Page 2-26

Service Request Registers

EIR	Error Service Request Register	0000 0020 _H	Page 2-28
SIR	Status Service Request Register	0000 0024 _H	Page 2-33
EILS	Error Service Request Line Select	0000 0028 _H	Page 2-38
SILS	Status Service Request Line Select	0000 002C _H	Page 2-42
EIES	Error Service Request Enable Set	0000 0030 _H	Page 2-46
EIER	Error Service Request Enable Reset	0000 0034 _H	Page 2-51
SIES	Status Service Request Enable Set	0000 0038 _H	Page 2-56
SIER	Status Service Request Enable Reset	0000 003C _H	Page 2-61
ILE	Service Request Line Enable	0000 0040 _H	Page 2-66
T0C	Timer 0 Configuration	0000 0044 _H	Page 2-67
T1C	Timer 1 Configuration	0000 0048 _H	Page 2-68
STPW1	Stop Watch Register1	0000 004C _H	Page 2-69
STPW2	Stop Watch Register 2	0000 0050 _H	Page 2-71
-	Reserved	0000 0054 _H - 0000 007C _H	-

Communication Controller Control Registers

SUCC1	SUC Configuration Register 1	0000 0080 _H	Page 2-72
SUCC2	SUC Configuration Register 2	0000 0084 _H	Page 2-80
SUCC3	SUC Configuration Register 3	0000 0088 _H	Page 2-81
NEMC	NEM Configuration Register	0000 008C _H	Page 2-82
PRTC1	PRT Configuration Register 1	0000 0090 _H	Page 2-83

FlexRay Protocol Controller (E-Ray)
Table 2-3 Registers OverviewE-Ray Kernel Registers (cont'd)

Register Short Name	Register Long Name	Address Offset	Details see
PRTC2	PRT Configuration Register 2	0000 0094 _H	Page 2-85
MHDC	MHD Configuration Register	0000 0098 _H	Page 2-86
-	Reserved	0000 009C _H	-
GTUC01	GTU Configuration Register 1	0000 00A0 _H	Page 2-87
GTUC02	GTU Configuration Register 2	0000 00A4 _H	Page 2-88
GTUC03	GTU Configuration Register 3	0000 00A8 _H	Page 2-89
GTUC04	GTU Configuration Register 4	0000 00AC _H	Page 2-90
GTUC05	GTU Configuration Register 5	0000 00B0 _H	Page 2-91
GTUC06	GTU Configuration Register 6	0000 00B4 _H	Page 2-92
GTUC07	GTU Configuration Register 7	0000 00B8 _H	Page 2-93
GTUC08	GTU Configuration Register 8	0000 00BC _H	Page 2-94
GTUC09	GTU Configuration Register 9	0000 00C0 _H	Page 2-95
GTUC10	GTU Configuration Register 10	0000 00C4 _H	Page 2-96
GTUC11	GTU Configuration Register 11	0000 00C8 _H	Page 2-97
-	Reserved	0000 00CC _H - 0000 00FC _H	-

Communication Controller Status Registers

CCSV	Communication Controller Status Vector	0000 0100 _H	Page 2-99
CCEV	Communication Controller Error Vector	0000 0104 _H	Page 2-104
-	Reserved	0000 0108 _H	-
-	Reserved	0000 010C _H	-
SCV	Slot Counter Value	0000 0110 _H	Page 2-105
MTCCV	Macrotick and Cycle Counter Value	0000 0114 _H	Page 2-106
RCV	Rate Correction Value	0000 0118 _H	Page 2-107
OCV	Offset Correction Value	0000 011C _H	Page 2-108
SFS	Sync Frame Status	0000 0120 _H	Page 2-109
SWNIT	Symbol Window and Network Idle Time Status	0000 0124 _H	Page 2-111
ACS	Aggregated Channel Status	0000 0128 _H	Page 2-114

FlexRay Protocol Controller (E-Ray)
Table 2-3 Registers OverviewE-Ray Kernel Registers (cont'd)

Register Short Name	Register Long Name	Address Offset	Details see
-	Reserved	0000 012C _H	-
ESIDnn	Even Sync ID Symbol Window nn	0000 0130 _H - 0000 0168 _H	Page 2-117
-	Reserved	0000 016C _H	-
OSIDnn	Odd Sync ID Symbol Window nn	0000 0170 - 0000 01A8 _H	Page 2-118
-	Reserved	0000 01AC _H	-
NMvN	Network Management Vector [1...3]	0000 01B0 _H - 0000 01B8 _H	Page 2-120
-	Reserved	0000 01BC _H - 0000 02FC _H	-

Message Buffer Control Registers

MRC	Message RAM Configuration	0000 0300 _H	Page 2-121
FRF	FIFO Rejection Filter	0000 0304 _H	Page 2-124
FRFM	FIFO Rejection Filter Mask	0000 0308 _H	Page 2-126
FCL	FIFO Critical Level	0000 030C _H	Page 2-127

Message Buffer Status Registers

MHDS	Message Handler Status	0000 0310 _H	Page 2-128
LDTS	Last Dynamic Transmit Slot	0000 0314 _H	Page 2-131
FSR	FIFO Status Register	0000 0318 _H	Page 2-132
MHDF	Message Handler Constraints Flags	0000 031C _H	Page 2-134
TXRQ1	Transmission Request Register 1	0000 0320 _H	Page 2-136
TXRQ2	Transmission Request Register 2	0000 0324 _H	Page 2-137
TXRQ3	Transmission Request Register 3	0000 0328 _H	Page 2-138
TXRQ4	Transmission Request Register 4	0000 032C _H	Page 2-139
NDAT1	New Data Register 1	0000 0330 _H	Page 2-140
NDAT2	New Data Register 2	0000 0334 _H	Page 2-141
NDAT3	New Data Register 3	0000 0338 _H	Page 2-142
NDAT4	New Data Register 4	0000 033C _H	Page 2-143
MBSC1	Message Buffer Status Changed 1	0000 0340 _H	Page 2-144
MBSC2	Message Buffer Status Changed 2	0000 0344 _H	Page 2-145

FlexRay Protocol Controller (E-Ray)

Table 2-3 Registers OverviewE-Ray Kernel Registers (cont'd)

Register Short Name	Register Long Name	Address Offset	Details see
MBSC3	Message Buffer Status Changed 3	0000 0348 _H	Page 2-146
MBSC4	Message Buffer Status Changed 4	0000 034C _H	Page 2-147
-	Reserved	0000 0350 _H - 0000 03 EC _H	-

Identification Registers

CREL	Core Release Registers	0000 03F0 _H	Page 2-148
ENDN	Endian Register	0000 03F4 _H	Page 2-150
-	Reserved	0000 03F6 _H - 0000 03FC _H	-

Input Buffer

WRDSn	Write Data Section [1...64]	0000 0400 _H - 0000 04FC _H	Page 2-151
WRHS1	Write Header Section 1	0000 0500 _H	Page 2-152
WRHS2	Write Header Section 2	0000 0504 _H	Page 2-155
WRHS3	Write Header Section 3	0000 0508 _H	Page 2-156
	Reserved	0000 050C _H	
IBCM	Input Buffer Command Mask	0000 0510 _H	Page 2-157
IBCR	Input Buffer Command Request	0000 0514 _H	Page 2-159
	Reserved	0000 0518 _H - 0000 05FC _H	

Output Buffer

RDDS _n	Read Data Section [1...64]	0000 0600 _H - 0000 06FC _H	Page 2-161
RDHS1	Read Header Section 1	0000 0700 _H	Page 2-162
RDHS2	Read Header Section 2	0000 0704 _H	Page 2-164
RDHS3	Read Header Section 3	0000 0708 _H	Page 2-166
MBS	Message Buffer Status	0000 070C _H	Page 2-168
OBCM	Output Buffer Command Mask	0000 0710 _H	Page 2-173
OBCR	Output Buffer Command Request	0000 0714 _H	Page 2-174
	Reserved	0000 0718 _H - 0000 07FC _H	-

2.4.2.1 Customer Registers

The address space from 0000_H to 000F_H is reserved for customer-specific registers.

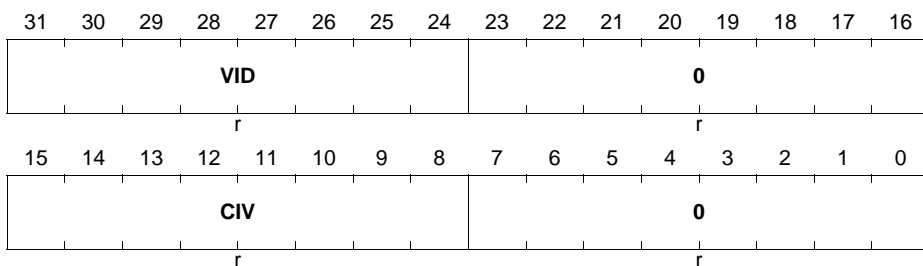
Product Identification Register (CUST0)

This register contains bit-fields about the product identification and is read only.

Note: Details on the Core Release will no longer be part of the CUST0 register and can be found in “CREL” on Page 2-148.

CUST0

Product Identification Register (0000_H) **Reset Value: C100 0100_H**



Field	Bits	Type	Description
CIV	[15:8]	r	Customer Interface Version
VID	[31:24]	r	Vendor ID Vendor ID uses the code identified by the JEDEC PUBLICATION: “Standard Manufacturer’s Identification Code, JEP106Q (Revision of JEP106P), JANUARY 2005, JEDEC SOLID STATE TECHNOLOGY ASSOCIATION”.
0	[23:16], [7:0]	r	Reserved Returns 0 if read; should be written with 0. <i>Note: Details on the Core Release will no longer be part of the CUST0 register and can be found in “CREL” on Page 2-148. Due to compatibility reasons, first Silicon of M1705 will have a reset value of CUST0 of 7_H.</i>

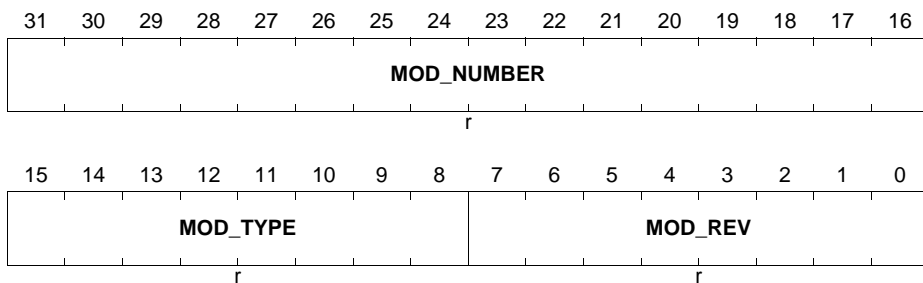
FlexRay Protocol Controller (E-Ray)

Module Identification Register (ID)

This register contains bit-fields identifying the E-Ray module in Infineons Module portfolio and is read only.

ID

Module Identification Register (0008_H) **Reset Value: 0044 C0xx_H**



Field	Bits	Type	Description
MOD_REV	[7:0]	r	Module Revision Number MOD_REV defines the module revision number. The value of a module revision starts with 01 _H (first revision).
MOD_TYPE	[15:8]	r	Module Type The value of this bit field is C0 _H . It defines the module as a 32-bit module.
MOD_NUMBER	[31:16]	r	Module Number Value This bit field defines a module identification number. For the E-Ray module the module identification number is 44 _H .

FlexRay Protocol Controller (E-Ray)

Busy Control Register (CUST1)

The Busy Control Register enables the automatic delay scheme. Furthermore it signals a timeout service request for the automatic delay scheme.

CUST1

Busy and Input Buffer Control Register

(0004_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0													IEN	OEN	INT0
rw													rw	rw	rwh

Field	Bits	Type	Description
INT0	0	rwh	CIF Timeout Service Request Status INT0 will be set if a timeout has occurred during the auto delay scheme and must be reset by writing zero to INT0. <i>Note: In case hardware sets INT0 and at the same point of time software clears INT0, INT0 is cleared.</i>
OEN	1	rw	Enable auto delay scheme for Output Buffer Control Register (OBCR) This control bit controls the delay scheme for Output Buffer Control Register (OBCR) read accesses. 0 _B Disable auto delay scheme for Output Buffer Control Register (OBCR) 1 _B Enable auto delay scheme for Output Buffer Control Register (OBCR)
IEN	2	rw	Enable auto delay scheme for Input Buffer Control Register (IBCR) This control bit controls the auto delay scheme for Input Buffer Control Register (IBCR) read accesses. 0 _B Disable auto delay scheme for Input Buffer Control Register (IBCR) 1 _B Enable auto delay scheme for Input Buffer Control Register (IBCR)

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
0	[31:3]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

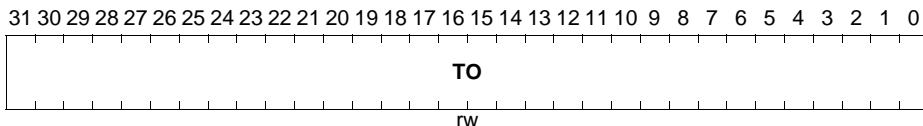
Customer Interface Timeout Counter Register (CUST3)

The Timeout Counter Register is realizing the timeout counter reload (startup) value for the automatic delay scheme (not the timeout down counter itself).

CUST3

Customer Interface Timeout Counter (000C_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
TO	[31:0]	rw	CIF Timeout Reload Value The 32-bit down counter reload (start-up) value must be setup for the automatic delay scheme.

Automatic Delayed Write Access to OBCR and IBCR.

Write and read accesses to the Output Buffer Control Register (**OBCR**) can be automatically stalled due to a ongoing transfer from the Message Buffer to the Output Buffer. Also write and read accesses to the Input Buffer Control Register (**IBCR**) may be automatically delayed due to a ongoing transfer from the Input Buffer to the Message Buffer.

This delay scheme can be controlled (enabled or disabled) by **CUST1.IEN** and **CUST1.OEN**. The maximum time to stall a write or read access is determined by a single timeout counter preloaded with the 32-bit value specified in the bitfield **CUST3.TO**. If the timeout counter counts down to zero before the transfer to/from the Message buffer is completed, the access (read or write) will be canceled and a service request will be generated. A canceled read access provides a 0 value. A canceled write access does not modify any bits in the **OBCR** or **IBCR**. In addition the bit **CUST1.INT0** of the service request status register will be set and must be reset by the host to disable the service request line.

The read and write access to the Output Buffer Control Register (**OBCR**) may be configured without automatic delay by clearing **CUST1.OEN** = 0_B. If setting **OBCR.REQ** = 1_B and immediately afterwards reading or writing **OBCR**, e.g. to set the **OBCR.VIEW** = 1_B, bit will lead to a canceled read or write operation, e.g. **OBCR.VIEW** remains cleared, and an error is signalled by a set **EIR.IOBA** = 1_B. Beside canceling the erroneous read or write operation, and setting the error bit, no further state change happens. So full operation is granted. **OBCR** remains read and write inaccessible until the transfer of data from the Message Buffer to the Output Buffer (MBF⇒OBF) is

FlexRay Protocol Controller (E-Ray)

completed. During this time span all read and write accesses to the Output Buffer Control Register (**OBCR**) are canceled. The transfer is completed when **OBCR.OBSYS** is cleared ($= 0_B$). Additionally a hardware signal `eray_tobc` may be used, e.g. for service request triggering, DMA triggering, or driving a pin, to communicate the access status.

The read and write access to the Output Buffer Control Register (**OBCR**) may be configured being automatic delayed by setting **CUST1.OEN** $= 1_B$ and configuring **CUST3.TO** to the maximum stall time acceptable to the system. If setting **OBCR.REQ** $= 1_B$ and immediately afterwards reading or writing to **OBCR**, e.g. to set the **OBCR.VIEW** $= 1_B$ bit, this read or write will be stalled until either the maximum delay time elapsed (in this case the read or write operation is cancelled after the stall time, e.g. **OBCR.VIEW** remains cleared, and an error is signalled by setting **EIR.IOBA** $= 1_B$) or the read or write completes normally, e.g. set **OBCR.VIEW** $= 1_B$, after the transfer of data from the Message Buffer to the Output Buffer (MBF \Rightarrow OBF) is finalized. During this time the bus is locked and no further access to E-Ray module is possible due to the ongoing stalled read or write operation. Because no access is possible to the E-Ray module, read or write stall may only be detected through the `eray_tobc` signal or due to other not processed read or write accesses to the E-Ray module.

The read and write access to the Input Buffer Control Register (**IBCR**) may also be configured without automatic delay by clearing **CUST1.IEN** $= 0_B$. If writing to **IBCR.IBRH** the input buffer are swapped (shadow IBF changes to host IBF and host IBF to shadow IBF), the content of the shadow IBF copied into the MBF (IBF \Rightarrow MBF), and **IBCR.IBSYS** $= 1_B$ set. If writing to **IBCR.IBRH** a second time while **IBCR.IBSYS** remained set ($= 1_B$: previous initiated copy process IBF \Rightarrow MBF ongoing) will correctly update **IBCR.IBRH** and set **IBCR.IBSYH** $= 1_B$. A third access, read or write, to **IBCR** while **IBCR.IBSYH** $= 1_B$ remains set ($= 1_B$), will lead to a canceled of this third access, read or write operation, to **IBCR** and an error is signalled by setting **EIR.IIBA** $= 1_B$. Beside canceling this last access to **IBCR**, read or write operation, and setting the error bit, no further state change happens. So full operation is granted. **IBCR** remains read and write inaccessible until the transfer of data from the Input Shadow Buffer to the Message Buffer (IBF \Rightarrow MBF) completed and once more the input buffers are swapped (shadow IBF changes to host IBF and host IBF to shadow IBF). During this time span all read and write accesses to the Input Buffer Control Register (**IBCR**) are canceled. The transfer is completed when **IBCR.IBSYH** is cleared ($= 0_B$). Additionally a hardware signal `eray_tibc` may be used, e.g. for service request triggering, DMA triggering, or driving a pin, to communicate the access status.

The read and write access to the Input Buffer Control Register (**IBCR**) may be configured being automatic delayed by setting **CUST1.IEN** $= 1_B$ and configuring **CUST3.TO** to the maximum stall time acceptable to the system. If writing to **IBCR.IBRH** the input buffer are swapped (shadow IBF changes to host IBF and host IBF to shadow IBF), the content of the shadow IBF copied into the MBF, and **IBCR.IBSYS** $= 1_B$ set. If writing to **IBCR.IBRH** a second time while **IBCR.IBSYS** remains set ($= 1_B$: previous initiated copy process ongoing) will correctly update **IBCR.IBRH** and set **IBCR.IBSYH** $= 1_B$. A third

FlexRay Protocol Controller (E-Ray)

access to **IBCR** while **IBCR.IBSYH** = 1_B remains set (= 1_B), will stall this read or write until either the maximum delay time elapsed (in this case the read or write operation is cancelled after the stall time and an error is signalled by setting **EIR.IOBA** = 1_B) or the read or write completes normally, after the transfer of data from the Input Shadow Buffer to the Message Buffer (IBF⇒MBF) is finalized and once more the input buffers are swapped (shadow IBF changes to host IBF and host IBF to shadow IBF). During this time the bus is locked and no further access to E-Ray module is possible due to the ongoing stalled read or write operation. Because no access is possible to the E-Ray module, read or write stall may only be detected through the `eray_tibc` signal or due to other not processed read or write accesses to the E-Ray module.

So if setting **CUST3.TO** = FFFFFFFF_H, **CUST1.IEN** = 1_B, and **CUST1.OEN** = 1_B will always grant a consistent data access of the host to the output and input buffers without the need of reading and taking the status of **OBCR.OBSYS** or **IBCR.IBSYH** into account. But this simplified access may cause system latencies and system performance loss.

2.4.2.2 Special Registers

Test Register 1 (TEST1)

The Test Register 1 holds the control bits to configure the test modes of the E-Ray module. Write access to these bits is only possible if bit **TEST1.WRTEN** is set to 1.

The Test Register 1 bits therefore can be used to test the interface to the physical layer (connectivity test) by driving / reading the respective pins.

When the E-Ray IP is operated in one of its test modes that requires **TEST1.WRTEN** to be set (RAM Test Mode, I/O Test Mode, Asynchronous Transmit Mode, and Loop Back Mode) only the selected test mode functionality is available.

The test functions are not available in addition to the normal operational mode functions, they change the functions of parts of the E-Ray module. Therefore normal operation as specified outside this chapter and as required by the FlexRay protocol specification and the FlexRay conformance test is not possible. Test mode functions may not be combined with each other or with FlexRay protocol functions.

The test mode features are intended for hardware testing or for FlexRay bus analyzer tools. They are not intended to be used in FlexRay applications

TEST1

Test Register 1

(0010_H)

Reset Value: 0000 0300_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CERB				CERA				0	TXE NB	TXE NA	TXB	TXA	RXB	RXA	
rh				rh				r	rwh	rwh	rwh	rwh	rh	rh	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				AOB		AOA	0	TMC			0	ELB E	WRT EN		
r				rh		rh	r	rw			r	rw	rw		

Field	Bits	Type	Description
WRTEN	0	rw	Write Test Register Enable Enables write access to the test registers. To set the bit from 0 to 1 the test mode key has to be written as defined on “Lock Register (LCK)” on Page 2-26 . The unlock sequence is not required when TEST1.WRTEN is kept at 1 while other bits of the register are changed. The bit can be reset to 0 at any time. 0 _B Write access to test registers disabled. 1 _B Write access to test registers enabled.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
ELBE	1	rw	External Loop Back Enable There are two possibilities to perform a loop back test. External loop back via physical layer or internal loop back for in-system self-test (default). In case of an internal loop back pins TXENA and TXENB are in their inactive state, pins TXDA and TXDB are set to HIGH, pins RXDA and RXDB are not evaluated. Bit ELBE is evaluated only when POC is in loop back mode and test multiplexer control is in non multiplexed mode TMC = 00. 0 _B Internal loop back (default) 1 _B External loop back
TMC	[5:4]	rw	Test Multiplexer Control 00 _B Normal signal path (default). 01 _B RAM Test Mode: Internal busses are multiplexed to make all RAM blocks of the E-Ray module directly accessible by the Host. This mode is intended to enable testing of the embedded RAM blocks during production testing. 10 _B I/O Test Mode: Output pins eray_txd1, eray_txd2, eray_txen1, eray_txen2 are driven to the values defined by bits TXA, TXB, TXENA, TXENB. The values applied to the input pins eray_rxd1, eray_rxd2 can be read from register bits RXA and RXB. 11 _B Reserved; should not be used.
AOA	8	rh	Activity on A The channel idle condition is specified in the FlexRay protocol spec v2.1, chapter 3, BITSTRB process (zChannelIdle). 0 _B No activity detected, channel A idle 1 _B Activity detected, channel A not idle
AOB	9	rh	Activity on B The channel idle condition is specified in the FlexRay protocol spec v2.1, chapter 3, BITSTRB process (zChannelIdle). 0 _B No activity detected, channel B idle 1 _B Activity detected, channel B not idle
RXA	16	rh	Read Channel A Receive Pin 0 _B eray_rxda = 0 1 _B eray_rxda = 1

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
RXB	17	rh	Read Channel B Receive Pin 0 _B eray_rxdb = 0 1 _B eray_rxdb = 1
TXA	18	rwh	Read or Write to Channel A Transmit Pin 0 _B eray_txda = 0 1 _B eray_txda = 1
TXB	19	rwh	Read or Write to Channel B Transmit Pin 0 _B eray_txdb = 0 1 _B eray_txdb = 1
TXENA	20	rwh	Read or Write to Channel A Transmit Enable Pin 0 _B eray_txena = 0 1 _B eray_txena = 1
TXENB	21	rwh	Read or Write to Channel B Transmit Enable Pin 0 _B eray_txenb = 0 1 _B eray_txenb = 1
CERA	[27:24]	rh	Coding Error Report Channel A¹⁾ Set when a coding error is detected on channel A. Reset to zero when register TEST1 is read or written. Once the CERA is set it will remain unchanged until the Host accesses the TEST1 register. 0000 _B No coding error detected 0001 _B Header CRC error detected 0010 _B Frame CRC error detected 0011 _B Frame Start Sequence FSS too long 0100 _B First bit of Byte Start Sequence BSS seen LOW 0101 _B Second bit of Byte Start Sequence BSS seen HIGH 0110 _B First bit of Frame End Sequence FES seen HIGH 0111 _B Second bit of Frame End Sequence FES seen LOW 1000 _B CAS / MTS symbol seen too short 1001 _B CAS / MTS symbol seen too long Other combinations are reserved.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
CERB	[31:28]	rh	Coding Error Report Channel B¹⁾ Set when a coding error is detected on channel B. Reset to zero when register TEST1 is read or written. Once the CERB is set it will remain unchanged until the Host accesses the TEST1 register. 0000 _B No coding error detected 0001 _B Header CRC error detected 0010 _B Frame CRC error detected 0011 _B Frame Start Sequence FSS too long 0100 _B First bit of Byte Start Sequence BSS seen LOW 0101 _B Second bit of Byte Start Sequence BSS seen HIGH 0110 _B First bit of Frame End Sequence FES seen HIGH 0111 _B Second bit of Frame End Sequence FES seen LOW 1000 _B CAS / MTS symbol seen too short 1001 _B CAS / MTS symbol seen too long Other combinations are reserved.
0	[3:2], [7:6], [15:10], [23:22]	r	Reserved Returns 0 if read; should be written with 0.

1) Coding errors are also signalled when the Communication Controller is in "MONITOR_MODE". The error codes regarding CAS / MTS symbols concern only the monitored bit pattern, irrelevant whether those bit patterns are seen in the symbol window or elsewhere.

Asynchronous Transmit Mode (ATM)

The asynchronous transmit mode is entered by writing 1110_B to the CHI Command Vector **SUCC1.CMD** in the SUC Configuration Register 1 (CHI command: ATM) while the Communication Controller is in "CONFIG" state and bit **TEST1.WRTEN** in the Test Register 1 is set to 1. This write operation has to be directly preceded by two consecutive write accesses to the Configuration Lock Key (unlock sequence). When called in any other state or when bit **TEST1.WRTEN** is not set, **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED". **CCSV.POCS** in the Communication Controller Status Vector will return 1110_B while the E-Ray module is in ATM mode. Asynchronous Transmit mode can be left by writing 0001_B (CHI command: "CONFIG") to the CHI Command Vector **SUCC1.CMD** in the SUC Configuration Register 1.

In ATM mode transmission of a FlexRay frame is triggered by writing the number of the respective message buffer to the Input Buffer Command Request register (**IBCR.IBRH**) while bit **IBCM.STXRS** in the Input Buffer Command Mask register is set to 1. In this mode wakeup, startup, and clock synchronization are bypassed. The CHI command SEND_MTS results in the immediate transmission of an MTS symbol.

FlexRay Protocol Controller (E-Ray)

The cycle counter value of frames send in ATM mode can be programmed via **MTCCV.CCV** (writeable in ATM and loop back mode only).

Loop Back Mode

The loop back mode is entered by writing 1111_B to the CHI Command Vector **SUCC1.CMD** in the SUC Configuration Register 1 (CHI command: LOOP_BACK) while the Communication Controller is in "CONFIG" state and bit **TEST1.WRTEN** in the Test Register 1 is set to 1. This write operation has to be directly preceded by two consecutive write accesses to the Configuration Lock Key (unlock sequence). When called in any other state or when bit **TEST1.WRTEN** is not set, **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED". **CCSV.POCS** in the Communication Controller Status Vector will show $0000\ 1101_H$ while the E-Ray module is in loop back mode.

Loop Back mode can be left by writing 0001_B (CHI command: "CONFIG") to the CHI Command Vector **SUCC1.CMD** in the SUC Configuration Register 1.

The loop back test mode is intended to check the module's internal data paths. Normal, time triggered operation is not possible in loop back mode.

There are two possibilities to perform a loop back test. External loop back via physical layer (**TEST1.ELBE** = 1) or internal loop back for in-system self-test (**TEST1.ELBE** = 0). In case of an internal loop back pins *eray_txen1,2_n* are in their inactive state, pins *eray_txd1,2* are set to HIGH, pins *eray_rxd1,2* are not evaluated.

When the Communication Controller is in loop back mode, a loop back test is started by the Host writing a message to the Input Buffer and requesting the transmission by writing to the Input Buffer Command Request register **IBCR**. The Message Handler will transfer the message into the Message RAM and then into the Transient Buffer of the selected channel. The Channel Protocol Controller (PRT) will read (in 32-bit words) the message from the transmit part of the Transient Buffer and load it into its Rx / Tx shift register. The serial transmission is looped back into the shift register; its content is written into the receive part of the channels's Transient Buffer before the next word is loaded.

The PRT and the Message Handler will then treat this transmitted message like a received message, perform an acceptance filtering on frame ID and receive channel, and store the message into the Message RAM if it passed acceptance filtering. The loop back test ends with the Host requesting this received message from the Message RAM and then checking the contents of the Output Buffer.

Each FlexRay channel is tested separately. The E-Ray cannot receive messages from the FlexRay bus while it is in the loop back mode.

The cycle counter value of frames used in loop back mode can be programmed via **MTCCV.CCV** (writeable in ATM and loop back mode only).

Note that in case of an odd payload the last two bytes of the looped-back payload will be shifted by 16 bits to the right inside the last 32-bit data word.

FlexRay Protocol Controller (E-Ray)

Test Register 2 (TEST2)

The Test Register 2 holds all bits required for the RAM test of the seven embedded RAM blocks of the E-Ray module. Write access to this register is only possible when **TEST1.WRTEN** in the Test Register 1 is set to 1.

TEST2

Test Register 2

(0014_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								0							
								r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDP B	WRP B									SSEL		0		RS	
rw	rw									rw		r		rw	

Field	Bits	Type	Description
RS	[2:0]	rw	RAM Select In RAM Test mode the RAM blocks selected by RS are mapped to module address 0000 0400 _H to 0000 07FF _H (1024 byte addresses). 000 _B Input Buffer RAM 1 (IBF1) 001 _B Input Buffer RAM 2 (IBF2) 010 _B Output Buffer RAM 1 (OBF1) 011 _B Output Buffer RAM 2 (OBF2) 100 _B Transient Buffer RAM A (TBF1) 101 _B Transient Buffer RAM B (TBF2) 110 _B Message RAM (MBF) 111 _B Reserved; should not be used.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SSEL	[6:4]	rw	Segment Select To enable access to the complete Message RAM (8192 byte addresses) the Message RAM is segmented. 000 _B access to RAM byte 0000 _H to 03FF _H enabled 001 _B access to RAM byte 0400 _H to 07FF _H enabled 010 _B access to RAM byte 0800 _H to 0BFF _H enabled 011 _B access to RAM byte 0C00 _H to 0FFF _H enabled 100 _B access to RAM byte 1000 _H to 11FF _H enabled 101 _B access to RAM byte 1400 _H to 17FF _H enabled 110 _B access to RAM byte 1800 _H to 1BFF _H enabled 111 _B access to RAM byte 1C00 _H to 1FFF _H enabled
WRPB	14	rw	Write Parity Bit Value of parity bit to be written to bit 32 of the addressed RAM word.
RDPB	15	rh	Read Parity Bit Value of parity bit read from bit 32 of the addressed RAM word.
0	3, [13:7], [31:16]	r	Reserved Returns 0 if read; should be written with 0.

RAM Test Mode

In RAM test mode (**TEST1.TMC** = 01_B), one of the seven RAM blocks can be selected for direct RD/WR access by programming **TEST2.RS**.

For external access the selected RAM block is mapped to address space 400_H to 7FF_H (1024 byte addresses or 256 word addresses).

Because the length of the Message RAM exceeds the available address space, the Message RAM is segmented into segments of 1024 byte. The segments can be selected by programming **TEST2.SSEL** in the Test Register 2.

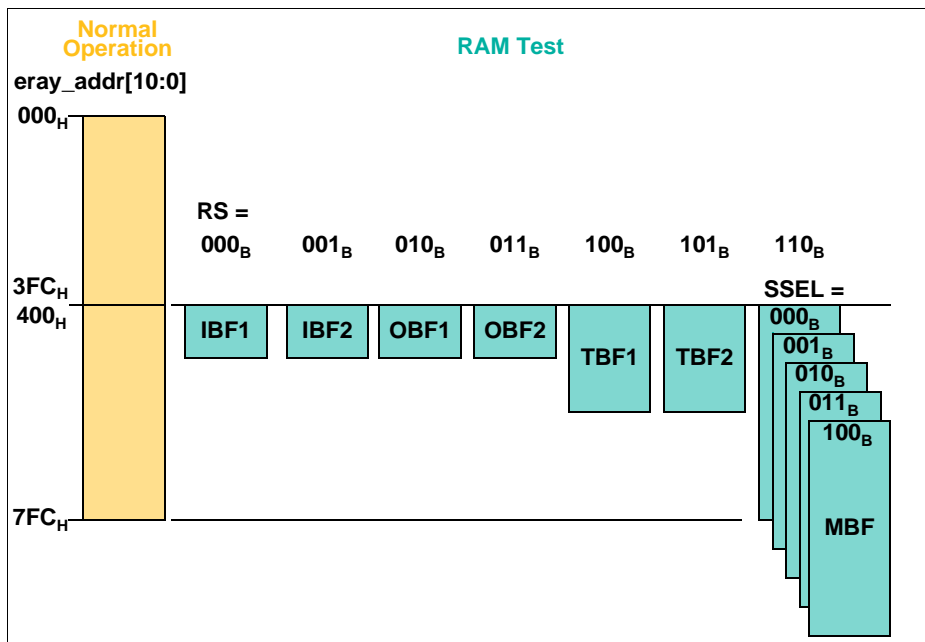


Figure 2-2 RAM test mode Access to E-Ray RAM Blocks

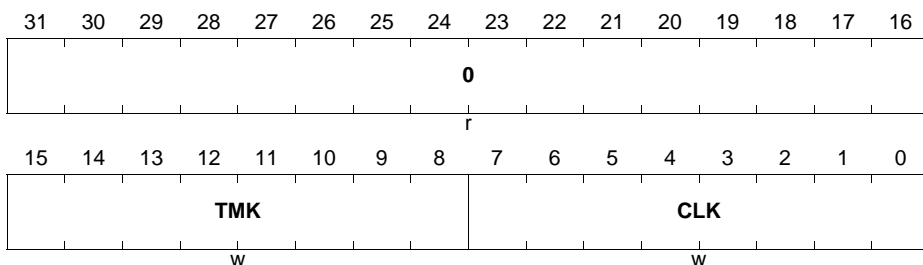
FlexRay Protocol Controller (E-Ray)

Lock Register (LCK)

The Lock Register is write-only. Reading the register will return 0000 0000_H.

LCK

Lock Register (001C_H) Reset Value: 0000 0000_H



Field	Bits	Type	Description
CLK	[7:0]	w	Configuration Lock Key To leave "CONFIG" state by writing to SUCC1.CMD commands READY, MONITOR_MODE, ATM, LOOP_BACK) in the SUC Configuration Register 1, the write operation has to be directly preceded by two consecutive write accesses to the Configuration Lock Key (unlock sequence). If the write sequence below is interrupted by other write accesses between the second write to the Configuration Lock Key and the write access to the SUCC1 register, the Communication Controller remains in "CONFIG" state and the sequence has to be repeated. First write: LCK.CLK = CE _H = 1100 1110 _B Second write: LCK.CLK = 31 _H = 0011 0001 _B Third write: SUCC1.CMD Returns 0 if read

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TMK	[15:8]	w	Test Mode Key To write bit TEST1.WRTEN in the Test Register to 1, the write operation has to be directly preceded by two consecutive write accesses to the Test Mode Key. If the write sequence is interrupted by other write accesses between the second write to the Test Mode Key and the write access to the Test1 register, bit TEST1.WRTEN is not set to 1 and the sequence has to be repeated. First write: LCK.TMK = $75_H = 0111\ 0101_B$ Second write: LCK.TMK = $8A_H = 1000\ 1010_B$ Second write: TEST1.WRTEN = 1 Returns 0 if read
0	[31:16]	r	Reserved Returns 0 if read; should be written with 0.

Note: In case the Host uses 8/16-bit accesses to write the listed bit fields, the programmer has to ensure that no “dummy accesses” e.g. to the remaining register bytes / words are inserted by the compiler.

To exit “CONFIG” state by writing to **SUCC1.CMD** in the SUC Configuration Register 1, the write operation has to be directly preceded by two consecutive write accesses to the Configuration Lock Key. If this write sequence is service requested by read accesses or write accesses to other locations, the Communication Controller remains in “CONFIG” state and the sequence has to be repeated.

First write: **LCK.CLK** = $CE_H = 1100\ 1110_B$

Second write: **LCK.CLK** = $31_H = 0011\ 0001_B$

2.4.2.3 Service Request Registers

The address space from 0020_H to 007F_H is reserved for service request registers.

Error Service Request Line Select (EILS)

The flags are set when the Communication Controller detects one of the listed error conditions. They remain set until the Host clears them. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect on the flag. A hard reset will also clear the register.

EIR

Error Service Request Register (0020_H) **Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0					TAB B	LTV B	EDB	0					TAB A	LTV A	EDA
r					rwh	rwh	rwh	r					rwh	rwh	rwh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0					MHF	IO BA	II BA	EFA	RFO	PER R	CCL	CCF	SFO	SFB M	PEM C
r					rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
PEMC	0	rwh	POC Error Mode Changed This flag is set whenever the error mode signalled by CCEV.ERRM in the Communication Controller Error Vector register has changed. 0 _B Error mode has not changed 1 _B Error mode has changed This flag is cleared by writing a 1.
CNA	1	rwh	Command Not Accepted The flag signals that the write access to the CHI command vector SUCC1.CMD in the SUC Configuration Register 1 was not successful because the requested command was not valid in the actual POC state, or because the CHI command was locked (CCL = 1). 0 _B CHI command accepted 1 _B CHI command not accepted This flag is cleared by writing a 1.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SFBM	2	rwh	<p>Sync Frames Below Minimum</p> <p>This flag signals that the number of sync frames received during the last communication cycle was below the limit required by the FlexRay protocol. May be set during startup and therefore should be cleared by the Host after the Communication Controller entered "NORMAL_ACTIVE" state.</p> <p>0_B Sync node: 1 or more sync frames received Non-sync node: 2 or more sync frames received</p> <p>1_B Less than the required minimum of sync frames received</p> <p>This flag is cleared by writing a 1.</p>
SFO	3	rwh	<p>Sync Frame Overflow</p> <p>Set when the number of sync frames received during the last communication cycle exceeds the maximum number of sync frames as defined by GTUC02.SNM in the GTU Configuration Register 2.</p> <p>0_B Number of received sync frames ≤ GTUC02.SNM</p> <p>1_B More sync frames received than configured by GTUC02.SNM</p> <p>This flag is cleared by writing a 1.</p>
CCF	4	rwh	<p>Clock Correction Failure</p> <p>This flag is set at the end of the cycle whenever one of the following errors occurred:</p> <ul style="list-style-type: none"> • Missing offset and / or rate correction • Clock Correction limit reached <p>The clock correction status is monitored in registers CCEV and SFS. A failure may occur during startup, therefore bit CCF should be cleared by the Host after the Communication Controller entered "NORMAL_ACTIVE" state.</p> <p>0_B Clock correction successful so far</p> <p>1_B Clock correction failed</p> <p>This flag is cleared by writing a 1.</p>

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
CCL	5	rwh	CHI Command Locked The flag signals that the write access to the CHI command vector SUCC1.CMD was not successful because the execution of the previous CHI command has not yet completed. In this case bit EIR.CNA is also set to 1. 0 _B CHI command accepted 1 _B CHI command not accepted This flag is cleared by writing a 1.
PERR	6	rh	Parity Error The flag signals a parity error to the Host. It is set whenever one of the flags MHDS.PIBF , MHDS.POBf , MHDS.PMR , MHDS.PTBF1 , MHDS.PTBF2 changes from 0 to 1. See also “Message Handler Status (MHDS)” on Page 2-128 . 0 _B No parity error detected 1 _B Parity error detected
RFO	7	rh	Receive FIFO Overrun The flag is set by the Communication Controller when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. The actual state of the FIFO is monitored in register FSR. 0 _B No receive FIFO overrun detected 1 _B A receive FIFO overrun has been detected
EFA	8	rwh	Empty FIFO Access This flag is set by the Communication Controller when the Host requests the transfer of a message from the receive FIFO via Output Buffer while the receive FIFO is empty. 0 _B No Host access to empty FIFO occurred 1 _B Host access to empty FIFO occurred

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
IIBA	9	rwh	<p>Illegal Input Buffer Access</p> <p>This flag is set by the Communication Controller when the Host wants to modify a message buffer via Input Buffer while the Communication Controller is not in "CONFIG" or "DEFAULT_CONFIG" state and one of the following conditions applies:</p> <ol style="list-style-type: none"> The Host writes to the Input Buffer Command Request register to modify the: <ul style="list-style-type: none"> Header section of message buffer 0, 1 if configured for transmission in key slot Header section of static message buffers with buffer number < MRC.FDB while MRC.SEC = 01_B Header section of any static or dynamic message buffer while MRC.SEC = 1x_B Header and / or data section of any message buffer belonging to the receive FIFO The Host writes to any register of the Input Buffer while IBCR.IBSYS is set to 1. <p>0_B No illegal Host access to Input Buffer occurred 1_B Illegal Host access to Input Buffer occurred</p>
IOBA	10	rwh	<p>Illegal Output Buffer Access</p> <p>This flag is set by the Communication Controller when the Host requests the transfer of a message buffer from the Message RAM to the Output Buffer while OBCR.OBSYS is set to 1.</p> <p>0_B No illegal Host access to Output Buffer occurred 1_B Illegal Host access to Output Buffer occurred</p>
MHF	11	rwh	<p>Message Handler Constraints Flag</p> <p>The flag signals a Message Handler constraints violation condition. It is set whenever one of the flags MHDF.SNUA, MHDF.SNUB, MHDF.FNFA, MHDF.FNFB, MHDF.TBFA, MHDF.TBFB, MHDF.WAHP changes from 0 to 1.</p> <p>0_B No Message Handler failure detected 1_B Message Handler failure detected</p>
EDA	16	rwh	<p>Error Detected on Channel A</p> <p>This bit is set whenever one of the flags ACS.SEDA, ACS.CEDA, ACS.CIA, ACS.SBVA changes from 0 to 1.</p> <p>0_B No error detected on channel A 1_B Error detected on channel A</p> <p>This flag is cleared by writing a 1.</p>

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
LTVA	17	rwh	Latest Transmit Violation Channel A The flag signals a latest transmit violation on channel A to the Host. 0_B No latest transmit violation detected on channel A 1_B Latest transmit violation detected on channel A This flag is cleared by writing a 1.
TABA	18	rwh	Transmission Across Boundary Channel A The flag signals to the Host that a transmission across a slot boundary occurred for channel A. 0_B No transmission across slot boundary detected on channel A 1_B Transmission across slot boundary detected on channel A This flag is cleared by writing a 1.
EDB	24	rwh	Error Detected on Channel B This bit is set whenever one of the flags ACS.SEDB , ACS.CEDB , ACS.CIB , ACS.SBVB changes from 0 to 1. 0_B No error detected on channel B 1_B Error detected on channel B This flag is cleared by writing a 1.
LTVB	25	rwh	Latest Transmit Violation Channel B The flag signals a latest transmit violation on channel B to the Host. 0_B No latest transmit violation detected on channel B 1_B Latest transmit violation detected on channel B This flag is cleared by writing a 1.
TABB	26	rwh	Transmission Across Boundary Channel B The flag signals to the Host that a transmission across a slot boundary occurred for channel B. 0_B No transmission across slot boundary detected on channel B 1_B Transmission across slot boundary detected on channel B This flag is cleared by writing a 1.
0	[15:12], [23:19], [31:27]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Status Service Request Register (SIR)

The flags are when the Communication Controller detects one of the listed events. The flags remain set until the Host clears them. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect on the flag. A hard reset will also clear the register.

SIR

Status Service Request Register (0024_H) **Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0						MTS B	WUP B	0						MTS A	WUP A
r						rwh	rwh	r						rwh	rwh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDS	MBS I	SUC S	SWE	TOB C	TIBC	TI1	TI0	NMV C	RF CL	RF NE	RXI	TXI	CYC S	CAS	WST
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rh	rh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
WST	0	rwh	Wakeup Status This flag is set when the wakeup status vector CCSV.WSV in the Communication Controller Status Vector register is modified by a protocol event. 0 _B Wakeup status unmodified 1 _B Wakeup status modified This flag is cleared by writing a 1.
CAS	1	rwh	Collision Avoidance Symbol This flag is set by the Communication Controller during STARTUP state when a CAS or potential CAS was received. 0 _B No bit pattern matching the CAS symbol received 1 _B Bit pattern matching the CAS symbol received This flag is cleared by writing a 1.
CYCS	2	rwh	Cycle Start Service Request This flag is set by the Communication Controller when a communication cycle starts 0 _B No communication cycle started 1 _B Communication cycle started This flag is cleared by writing a 1.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TXI	3	rwh	Transmit Service Request This flag is set by the Communication Controller at the end of frame transmission if bit WRHS1.MBI in the respective message buffer is set to 1 (see Table 2-24). 0 _B No frame transmitted from a transmit buffer with WRHS1.MBI = 1 1 _B At least one frame was transmitted from a transmit buffer with WRHS1.MBI = 1 This flag is cleared by writing a 1.
RXI	4	rwh	Receive Service Request This flag is set by the Communication Controller whenever the set condition of a message buffer ND flag is fulfilled (see Section to Section) and if bit WRHS1.MBI of that message buffer is set to 1 (see Table 2-24). 0 _B No ND flag of a receive buffer with WRHS1.MBI = 1 has been set to 1 1 _B At least one ND flag of a receive buffer with WRHS1.MBI = 1 has been set to 1 This flag is cleared by writing a 1.
RFNE	5	rh	Receive FIFO Not Empty This flag is set by the Communication Controller when a received valid frame was stored into the empty receive FIFO.m The actual state of the receive FIFO is monitored in register FSR 0 _B Receive FIFO is empty 1 _B Receive FIFO is not empty
RFCL	6	rh	Receive FIFO Critical Level This flag is set when a valid receive FIFO fill level FSR.RFFL is equal or greater than the critical level as configured by FCL.CL . 0 _B Receive FIFO below critical level 1 _B Receive FIFO critical level reached
NMVC	7	rwh	Network Management Vector Changed This service request flag signals a change in the Network Management Vector visible to the Host. 0 _B No change in the network management vector 1 _B Network management vector changed This flag is cleared by writing a 1.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TIO	8	rwh	<p>Timer Service Request 0</p> <p>This flag is set whenever timer 0 matches the conditions configured in the Timer Service Request 0 Configuration Register TOC. A Timer Service Request 0 is also signalled on pin eray_tint0.</p> <p>0_B No Timer Service Request 0 1_B Timer Service Request 0 occurred This flag is cleared by writing a 1.</p>
TI1	9	rwh	<p>Timer Service Request 1</p> <p>This flag is set whenever the conditions programmed in the Timer Service Request 1 Configuration Register T1C are met. A Timer Service Request 1 is also signalled on pin eray_tint1.</p> <p>0_B No Timer Service Request 1 1_B Timer Service Request 1 occurred This flag is cleared by writing a 1.</p>
TIBC	10	rwh	<p>Transfer Input Buffer Completed</p> <p>This flag is set whenever a transfer from Input Buffer to the Message RAM has completed and bit IBCR.IBSYS in the Input Buffer Command Request register has been reset by the Message Handler.</p> <p>0_B No transfer completed 1_B Transfer between Input Buffer and Message RAM completed This flag is cleared by writing a 1.</p>
TOBC	11	rwh	<p>Transfer Output Buffer Completed</p> <p>This flag is set whenever a transfer from Message RAM to the Output Buffer has completed and bit OBCR.OBSYS in the Output Buffer Command Request register has been reset by the Message Handler.</p> <p>0_B No transfer completed 1_B Transfer between Message RAM and the Output Buffer completed This flag is cleared by writing a 1.</p>

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SWE	12	rwh	Stop Watch Event This flag is set after a stop watch activation when the actual cycle counter and macrotick value are stored in the Stop Watch Register 1 (STPW1) (see “ Stop Watch Register 1 (STPW1) ” on Page 2-69). 0 _B No Stop Watch Event 1 _B Stop Watch Event occurred This flag is cleared by writing a 1.
SUCS	13	rwh	Startup Completed Successfully This flag is set whenever a startup completed successfully and the Communication Controller entered “NORMAL_ACTIVE” state. 0 _B No startup completed successfully 1 _B Startup completed successfully This flag is cleared by writing a 1.
MBSI	14	rwh	Message Buffer Status Service Request This flag is set by the Communication Controller when the message buffer status MBS has changed and if bit RDHS1.MBI of that message buffer is set (see Table 2-24). 0 _B No message buffer status change of message buffer with RDHS1.MBI = 1 has changed 1 _B Message buffer status of at least one message buffer with RDHS1.MBI = 1 has changed This flag is cleared by writing a 1.
SDS	15	rwh	Start of Dynamic Segment This flag is set by the Communication Controller when the dynamic segment starts. 0 _B Dynamic segment not yet started 1 _B Dynamic segment started
WUPA	16	rwh	Wakeup Pattern Channel A This flag is set by the Communication Controller when a wakeup pattern was received on channel A. Only set when the Communication Controller is in “WAKEUP”, “READY”, or “STARTUP” state, or when in Monitor mode. 0 _B No wakeup pattern received on channel A 1 _B Wakeup pattern received on channel A This flag is cleared by writing a 1.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
MTSA	17	rwh	MTS Received on Channel A (vSS!ValidMTSA) Media Access Test symbol received on channel A during the proceeding symbol window. Updated by the Communication Controller for each channel at the end of the symbol window. 0 _B No MTS symbol received on channel A 1 _B MTS symbol received on channel A This flag is cleared by writing a 1.
WUPB	24	rwh	Wakeup Pattern Channel B This flag is set by the Communication Controller when a wakeup pattern was received on channel B. Only set when the Communication Controller is in "WAKEUP", "READY", or "STARTUP" state, or when in Monitor mode. 0 _B No wakeup pattern received on channel B 1 _B Wakeup pattern received on channel B This flag is cleared by writing a 1.
MTSB	25	rwh	MTS Received on Channel B (vSS!ValidMTSB) Media Access Test symbol received on channel B during the proceeding symbol window. Updated by the Communication Controller for each channel at the end of the symbol window. 0 _B No MTS symbol received on channel B 1 _B MTS symbol received on channel B This flag is cleared by writing a 1.
0	[23:18], [31:26]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SFOL	3	rw	Sync Frame Overflow Service Request Line 0_B Service Request assigned to service request line eray_int0 1_B Service Request assigned to service request line eray_int1
CCFL	4	rw	Clock Correction Failure Service Request Line 0_B Service Request assigned to service request line eray_int0 1_B Service Request assigned to service request line eray_int1
CCLL	5	rw	CHI Command Locked Service Request Line 0_B Service Request assigned to service request line eray_int0 1_B Service Request assigned to service request line eray_int1
PERRL	6	rw	Parity Error Service Request Line 0_B Service Request assigned to service request line eray_int0 1_B Service Request assigned to service request line eray_int1
RFOL	7	rw	Receive FIFO Overrun Service Request Line 0_B Service Request assigned to service request line eray_int0 1_B Service Request assigned to service request line eray_int1
EFAL	8	rw	Empty FIFO Access Service Request Line 0_B Service Request assigned to service request line eray_int0 1_B Service Request assigned to service request line eray_int1
IIBAL	9	rw	Illegal Input Buffer Access Service Request Line 0_B Service Request assigned to service request line eray_int0 1_B Service Request assigned to service request line eray_int1

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
IOBAL	10	rw	Illegal Output Buffer Access Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
MHFL	11	rw	Message Handler Constrains Flag Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
EDAL	16	rw	Error Detected on Channel A Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
LTVAL	17	rw	Latest Transmit Violation Channel A Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int
TABAL	18	rw	Transmission Across Boundary Channel A Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
EDBL	24	rw	Error Detected on Channel B Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
LTVBL	25	rw	Latest Transmit Violation Channel B Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TABBL	26	rw	Transmission Across Boundary Channel A Service Request Line 0_B Service Request assigned to service request line eray_int0 1_B Service Request assigned to service request line eray_int1
0	[15:12], [23:19], [31:27]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Status Service Request Line Select (SILS)

The Status Service Request Line Select register assign an service request generated by a specific status service request flag from register [SIR](#) to one of the two module service request lines eray_int0 or eray_int1:

0 = Interrupt assigned to interrupt line eray_int0

1 = Interrupt assigned to interrupt line eray_int1

SILS

Status Service Request Line Select (002C_H)

Reset Value: 0303 FFFF_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			0			MTS BL	WUP BL				0			MTS AL	WUP AL
			r			rw	rw				r			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDS L	MBS IL	SUC SL	SWE L	TOB CL	TIBC L	TI1L	TI0L	NMV CL	RFC LL	RFN EL	RXIL	TXIL	CYC SL	CAS L	WST L
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
WSTL	0	rw	Wakeup Status Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
CASL	1	rw	Collision Avoidance Symbol Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
CYCSL	2	rw	Cycle Start Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TXIL	3	rw	Transmit Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
RXIL	4	rw	Receive Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
RFNEL	5	rw	Receive FIFO Not Empty Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
RFCLL	6	rw	Receive FIFO Critical Level Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
NMVCL	7	rw	Network Management Vector Changed Service Request Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int01
TI0L	8	rw	Timer Service Request 0 Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
TI1L	9	rw	Timer Service Request 1 Line
			0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TIBCL	10	rw	Transfer Input Buffer Completed Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
TOBCL	11	rw	Transfer Output Buffer Completed Service Request Line 1 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
SWEL	12	rw	Stop Watch Event Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
SUCSL	13	rw	Startup Completed Successfully Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
MBSIL	14	rw	Message Buffer Status Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
SDSL	15	rw	Start of Dynamic Segment Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
WUPAL	16	rw	Wakeup Pattern Channel A Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
MTSAL	17	rw	Media Access Test Symbol Channel A Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
WUPBL	24	rw	Wakeup Pattern Channel B Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
MTSBL	25	rw	Media Access Test Symbol Channel B Service Request Line 0 _B Service Request assigned to service request line eray_int0 1 _B Service Request assigned to service request line eray_int1
0	[23:18], [31:26]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Error Service Request Enable Set (EIES)

The settings in the Error Service Request Enable register determine which status changes in the Error Service Request Register will result in a service request. The enable bits are set by writing to EIES and reset by writing to EIER. Writing a 1 sets the specific enable bit, a 0 has no effect.

EIES

Error Service Request Enable Set (0030_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					TAB	LTV	EDB						TAB	LTV	EDA
					BE	BE	E						AE	AE	E
					rwh	rwh	rwh						rwh	rwh	rwh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					MHF	IOB	IIBA	EFA	RFO	PER	CCL	CCF	SFO	SFB	CNA
					E	AE	E	E	E	RE	E	E	E	ME	CE
					rwh	rwh	rwh	rwh	rwh	rwh	r	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
PEMCE	0	rwh	POC Error Mode Changed Service Request Enable Read: 0 _B Protocol Error Mode Changed Service Request disabled 1 _B Protocol Error Mode Changed Service Request enabled Write: 0 _B Unchanged 1 _B Enable Protocol Error Mode Changed Service Request
CNAE	1	rwh	Command Not Accepted Service Request Enable Read: 0 _B Command Not Valid Service Request disabled 1 _B Command Not Valid Service Request enabled Write: 0 _B Unchanged 1 _B Enable Command Not Valid Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SFBME	2	rwh	Sync Frames Below Minimum Service Request Enable Read: 0 _B Sync Frames Below Minimum Service Request disabled 1 _B Sync Frames Below Minimum Service Request enabled Write: 0 _B Unchanged 1 _B Enable Sync Frames Below Minimum Service Request
SFOE	3	rwh	Sync Frame Overflow Service Request Enable Read: 0 _B Sync Frame Overflow Service Request disabled 1 _B Sync Frame Overflow Service Request enabled Write: 0 _B Unchanged 1 _B Enable Protocol Error Mode Changed Service Request
CCFE	4	rwh	Clock Correction Failure Service Request Enable Read: 0 _B Clock Correction Failure Service Request disabled 1 _B Clock Correction Failure Service Request enabled Write: 0 _B Unchanged 1 _B Enable Clock Correction Failure Service Request
CCLE	5	rwh	CHI Command Locked Service Request Enable Read: 0 _B CHI Command Locked Service Request disabled 1 _B CHI Command Locked Service Request enabled Write: 0 _B Unchanged 1 _B Enable CHI Command Locked Service Request
PERRE	6	rwh	Parity Error Service Request Enable Read: 0 _B Parity Error Service Request disabled 1 _B Parity Error Service Request enabled Write: 0 _B Unchanged 1 _B Enable Parity Error Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
RFOE	7	rwh	Receive FIFO Overrun Service Request Enable Read: 0 _B Receive FIFO Overrun Service Request disabled 1 _B Receive FIFO Overrun Service Request enabled Write: 0 _B Unchanged 1 _B Enable Receive FIFO Overrun Service Request
EFAE	8	rwh	Empty FIFO Access Service Request Enable Read: 0 _B Empty FIFO Access Service Request disabled 1 _B Empty FIFO Access Service Request enabled Write: 0 _B Unchanged 1 _B Enable Empty FIFO Access Service Request
IIBAE	9	rwh	Illegal Input Buffer Access Service Request Enable Read: 0 _B Illegal Input Buffer Access Service Request disabled 1 _B Illegal Input Buffer Access Service Request enabled Write: 0 _B Unchanged 1 _B Enable Illegal Input Buffer Access Service Request
IOBAE	10	rwh	Illegal Output Buffer Access Service Request Enable Read: 0 _B Illegal Output Buffer Access Service Request disabled 1 _B Illegal Output Buffer Access Service Request enabled Write: 0 _B Unchanged 1 _B Enable Illegal Output Buffer Access Service Request
MHFE	11	rwh	Message Handler Constraints Flag Service Request Enable Read: 0 _B Message Handler Constraints Flag Service Request disabled 1 _B Message Handler Constraints Flag Service Request enabled Write: 0 _B Unchanged 1 _B Enable Message Handler Constraints Flag Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
EDAE	16	rwh	Error Detected on Channel A Service Request Enable Read: 0 _B Error Detected on Channel A Service Request disabled 1 _B Error Detected on Channel A Service Request enabled Write: 0 _B Unchanged 1 _B Enable Error Detected on Channel A Service Request
LTVAE	17	rwh	Latest Transmit Violation Channel A Service Request Enable Read: 0 _B Latest Transmit Violation Channel A Service Request disabled 1 _B Latest Transmit Violation Channel A Service Request enabled Write: 0 _B Unchanged 1 _B Enable Latest Transmit Violation Channel A Service Request
TABAE	18	rwh	Transmission Across Boundary Channel A Service Request Enable Read: 0 _B Transmission Across Boundary Channel A Service Request disabled 1 _B Transmission Across Boundary Channel A Service Request enabled Write: 0 _B Unchanged 1 _B Enable Transmission Across Boundary Channel A Service Request
EDBE	24	rwh	Error Detected on Channel B Service Request Enable Read: 0 _B Error Detected on Channel B Service Request disabled 1 _B Error Detected on Channel B Service Request enabled Write: 0 _B Unchanged 1 _B Enable Error Detected on Channel B Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
LTVBE	25	rwh	Latest Transmit Violation Channel B Service Request Enable Read: 0 _B Latest Transmit Violation Channel B Service Request disabled 1 _B Latest Transmit Violation Channel B Service Request enabled Write: 0 _B Unchanged 1 _B Enable Latest Transmit Violation Channel B Service Request
TABBE	26	rwh	Transmission Across Boundary Channel B Service Request Enable Read: 0 _B Transmission Across Boundary Channel B Service Request disabled 1 _B Transmission Across Boundary Channel B Service Request enabled Write: 0 _B Unchanged 1 _B Enable Transmission Across Boundary Channel B Service Request
0	[15:12], [23:19], [31:27]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SFBME	2	rw	Sync Frames Below Minimum Service Request Enable Read: 0 _B Sync Frames Below Minimum Service Request disabled 1 _B Sync Frames Below Minimum Service Request enabled Write: 0 _B Unchanged 1 _B Disable Sync Frames Below Minimum Service Request
SFOE	3	rw	Sync Frame Overflow Service Request Enable Read: 0 _B Sync Frame Overflow Service Request disabled 1 _B Sync Frame Overflow Service Request enabled Write: 0 _B Unchanged 1 _B Disable Protocol Error Mode Changed Service Request
CCFE	4	rw	Clock Correction Failure Service Request Enable Read: 0 _B Clock Correction Failure Service Request disabled 1 _B Clock Correction Failure Service Request enabled Write: 0 _B Unchanged 1 _B Disable Clock Correction Failure Service Request
CCLE	5	rwh	CHI Command Locked Service Request Enable Read: 0 _B CHI Command Locked Service Request disabled 1 _B CHI Command Locked Service Request enabled Write: 0 _B Unchanged 1 _B Disable CHI Command Locked Service Request
PERRE	6	rw	Parity Error Service Request Enable Read: 0 _B Parity Error Service Request disabled 1 _B Parity Error Service Request enabled Write: 0 _B Unchanged 1 _B Disable Parity Error Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
RFOE	7	rw	Receive FIFO Overrun Service Request Enable Read: 0 _B Receive FIFO Overrun Service Request disabled 1 _B Receive FIFO Overrun Service Request enabled Write: 0 _B Unchanged 1 _B Disable Receive FIFO Overrun Service Request
EFAE	8	rwh	Empty FIFO Access Service Request Enable Read: 0 _B Empty FIFO Access Service Request disabled 1 _B Empty FIFO Access Service Request enabled Write: 0 _B Unchanged 1 _B Disable Empty FIFO Access Service Request
IIBAE	9	rwh	Illegal Input Buffer Access Service Request Enable Read: 0 _B Illegal Input Buffer Access Service Request disabled 1 _B Illegal Input Buffer Access Service Request enabled Write: 0 _B Unchanged 1 _B Disable Illegal Input Buffer Access Service Request
IOBAE	10	rwh	Illegal Output Buffer Access Service Request Enable Read: 0 _B Illegal Output Buffer Access Service Request disabled 1 _B Illegal Output Buffer Access Service Request enabled Write: 0 _B Unchanged 1 _B Disable Illegal Output Buffer Access Service Request
MHFE	11	rwh	Message Handler Constraints Flag Service Request Enable Read: 0 _B Message Handler Constraints Flag Service Request disabled 1 _B Message Handler Constraints Flag Service Request enabled Write: 0 _B Unchanged 1 _B Disable Message Handler Constraints Flag Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
EDAE	16	rw	Error Detected on Channel A Service Request Enable Read: 0 _B Error Detected on Channel A Service Request disabled 1 _B Error Detected on Channel A Service Request enabled Write: 0 _B Unchanged 1 _B Disable Error Detected on Channel A Service Request
LTVAE	17	rw	Latest Transmit Violation Channel A Service Request Enable Read: 0 _B Latest Transmit Violation Channel A Service Request disabled 1 _B Latest Transmit Violation Channel A Service Request enabled Write: 0 _B Unchanged 1 _B Disable Latest Transmit Violation Channel A Service Request
TABAE	18	rwh	Transmission Across Boundary Channel A Service Request Enable Read: 0 _B Transmission Across Boundary Channel A Service Request disabled 1 _B Transmission Across Boundary Channel A Service Request enabled Write: 0 _B Unchanged 1 _B Enable Transmission Across Boundary Channel A Service Request
EDBE	24	rw	Error Detected on Channel B Service Request Enable Read: 0 _B Error Detected on Channel B Service Request disabled 1 _B Error Detected on Channel B Service Request enabled Write: 0 _B Unchanged 1 _B Disable Error Detected on Channel B Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
LTVBE	25	rw	Latest Transmit Violation Channel B Service Request Enable Read: 0 _B Latest Transmit Violation Channel B Service Request disabled 1 _B Latest Transmit Violation Channel B Service Request enabled Write: 0 _B Unchanged 1 _B Disable Latest Transmit Violation Channel B Service Request
TABBE	26	rwh	Transmission Across Boundary Channel B Service Request Enable Read: 0 _B Transmission Across Boundary Channel B Service Request disabled 1 _B Transmission Across Boundary Channel B Service Request enabled Write: 0 _B Unchanged 1 _B Disable Transmission Across Boundary Channel B Service Request
0	[15:12], [23:19], [31:27]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Status Service Request Enable Set (SIES)

The settings in the Status Service Request Enable Set register determine which status changes in the Status Service Request Register will result in a service request. The enable bits are set by writing to SIES and reset by writing to SIER. Writing a 1 sets the specific enable bit, a 0 has no effect.

SIES

Status Service Request Enable Set (0038_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0						MTS BE	WUP BE	0						MTS AE	WUP AE
r						rw	rw	r						rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDS E	MBS IE	SUC SE	SWE E	TOB CE	TIBC E	TI1E	TI0E	NMV CE	RFC LE	RFN EE	RXIE	TXIE	CYC SE	CAS E	WST E
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
WSTE	0	rw	Wakeup Status Service Request Enable Read: 0 _B Wakeup Status Service Request disabled 1 _B Wakeup Status Service Request enabled Write: 0 _B Unchanged 1 _B Enable Wakeup Status Service Request
CASE	1	rw	Collision Avoidance Symbol Service Request Enable Read: 0 _B Collision Avoidance Symbol Service Request disabled 1 _B Collision Avoidance Symbol Service Request enabled Write: 0 _B Unchanged 1 _B Enable Collision Avoidance Symbol Service Request
CYCSE	2	rw	Cycle Start Service Request Enable Read: 0 _B Cycle Start Service Request disabled 1 _B Cycle Start Service Request enabled Write: 0 _B Unchanged 1 _B Enable Cycle Start Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TXIE	3	rw	Transmit Service Request Enable Read: 0 _B Transmit Service Request disabled 1 _B Transmit Service Request enabled Write: 0 _B Unchanged 1 _B Enable Transmit Service Request
RXIE	4	rw	Receive Service Request Enable Read: 0 _B Receive Service Request disabled 1 _B Receive Service Request enabled Write: 0 _B Unchanged 1 _B Enable Receive Service Request
RFNEE	5	rw	Receive FIFO Not Empty Service Request Enable Read: 0 _B Receive FIFO Not Empty Service Request disabled 1 _B Receive FIFO Not Empty Service Request enabled Write: 0 _B Unchanged 1 _B Enable Receive FIFO Not Empty Service Request
RFCLE	6	rw	Receive FIFO Critical Level Service Request Enable Read: 0 _B Receive FIFO Critical Level Service Request disabled 1 _B Receive FIFO Critical Level Service Request enabled Write: 0 _B Unchanged 1 _B Enable Receive FIFO Critical Level Service Request
NMVCE	7	rw	Network Management Vector Changed Service Request Enable Read: 0 _B Network Management Vector Changed Service Request disabled 1 _B Network Management Vector Changed Service Request enabled Write: 0 _B Unchanged 1 _B Enable Network Management Vector Changed Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TI0E	8	rw	Timer Service Request 0 Enable Read: 0 _B Timer Service Request 0 disabled 1 _B Timer Service Request 0 enabled Write: 0 _B Unchanged 1 _B Enable Timer Service Request 0
TI1E	9	rw	Timer Service Request 1 Enable Read: 0 _B Timer Service Request 1 disabled 1 _B Timer Service Request 1 enabled Write: 0 _B Unchanged 1 _B Enable Timer Service Request 1
TIBCE	10	rw	Transfer Input Buffer Completed Service Request Enable Read: 0 _B Wakeup Status Service Request disabled 1 _B Wakeup Status Service Request enabled Write: 0 _B Unchanged 1 _B Enable Wakeup Status Service Request
TOBCE	11	rw	Transfer Output Buffer Completed Service Request Enable Read: 0 _B Transfer Input Buffer Completed Service Request disabled 1 _B Transfer Input Buffer Completed Service Request enabled Write: 0 _B Unchanged 1 _B Enable Transfer Input Buffer Completed Service Request
SWEE	12	rw	Stop Watch Event Service Request Enable Read: 0 _B Stop Watch Event Service Request disabled 1 _B Stop Watch Event Service Request enabled Write: 0 _B Unchanged 1 _B Enable Stop Watch Event Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SUCSE	13	rw	Startup Completed Successfully Service Request Enable Read: 0 _B Startup Completed Successfully Service Request disabled 1 _B Startup Completed Successfully Service Request enabled Write: 0 _B Unchanged 1 _B Enable Startup Completed Successfully Service Request
MBSIE	14	rw	Message Buffer Status Service Request Enable Read: 0 _B Message Buffer Status Service Request disabled 1 _B Message Buffer Status Service Request enabled Write: 0 _B Unchanged 1 _B Enable Message Buffer Status Service Request
SDSE	15	rw	Start of Dynamic Segment Service Request Enable Read: 0 _B Start of Dynamic Service Request disabled 1 _B Start of Dynamic Service Request enabled Write: 0 _B Unchanged 1 _B Enable Start of Dynamic Service Request
WUPAE	16	rw	Wakeup Pattern Channel A Service Request Enable Read: 0 _B Wakeup Pattern Channel A Service Request disabled 1 _B Wakeup Pattern Channel A Service Request enabled Write: 0 _B Unchanged 1 _B Enable Wakeup Pattern Channel A Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
MTSAE	17	rw	Media Access Test Symbol Channel A Service Request Enable Read: 0 _B Media Access Test Symbol Channel A Service Request disabled 1 _B Media Access Test Symbol Channel A Service Request enabled Write: 0 _B Unchanged 1 _B Enable Media Access Test Symbol Channel A Service Request
WUPBE	24	rw	Wakeup Pattern Channel B Service Request Enable Read: 0 _B Wakeup Pattern Channel B Service Request disabled 1 _B Wakeup Pattern Channel B Service Request enabled Write: 0 _B Unchanged 1 _B Enable Wakeup Pattern Channel A Service Request
MTSBE	25	rw	Media Access Test Symbol Channel B Service Request Enable Read: 0 _B Media Access Test Symbol Channel B Service Request disabled 1 _B Media Access Test Symbol Channel B Service Request enabled Write: 0 _B Unchanged 1 _B Enable Media Access Test Symbol Channel B Service Request
0	[23:18], [31:26]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Status Service Request Enable Reset (SIER)

The settings in the Status Service Request Enable Reset register determine which status changes in the Status Service Request Register will result in a service request. The enable bits are set by writing to SIES and reset by writing to SIER. Writing a 1 resets the specific enable bit, a 0 has no effect.

SIER

Status Service Request Enable Reset(003C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0						MTS BE	WUP BE	0						MTS AE	WUP AE
r						rw	rw	r						rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDS E	MBS IE	SUC SE	SWE E	TOB CE	TIBC E	TI1E	TI0E	NMV CE	RFC LE	RFN EE	RXIE	TXIE	CYC SE	CAS E	WST E
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
WSTE	0	rw	Wakeup Status Service Request Enable Read: 0 _B Wakeup Status Service Request disabled 1 _B Wakeup Status Service Request enabled Write: 0 _B Unchanged 1 _B Disable Wakeup Status Service Request
CASE	1	rw	Collision Avoidance Symbol Service Request Enable Read: 0 _B Collision Avoidance Symbol Service Request disabled 1 _B Collision Avoidance Symbol Service Request enabled Write: 0 _B Unchanged 1 _B Disable Collision Avoidance Symbol Service Request
CYCSE	2	rw	Cycle Start Service Request Enable Read: 0 _B Cycle Start Service Request disabled 1 _B Cycle Start Service Request enabled Write: 0 _B Unchanged 1 _B Disable Cycle Start Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TXIE	3	rw	Transmit Service Request Enable Read: 0 _B Transmit Service Request disabled 1 _B Transmit Service Request enabled Write: 0 _B Unchanged 1 _B Disable Transmit Service Request
RXIE	4	rw	Receive Service Request Enable Read: 0 _B Receive Service Request disabled 1 _B Receive Service Request enabled Write: 0 _B Unchanged 1 _B Disable Receive Service Request
RFNEE	5	rw	Receive FIFO Not Empty Service Request Enable Read: 0 _B Receive FIFO Not Empty Service Request disabled 1 _B Receive FIFO Not Empty Service Request enabled Write: 0 _B Unchanged 1 _B Disable Receive FIFO Not Empty Service Request
RFCLE	6	rw	Receive FIFO Critical Level Service Request Enable Read: 0 _B Service Request disabled 1 _B Receive FIFO Critical Level Service Request enabled Write: 0 _B Unchanged 1 _B Disable Receive FIFO Critical Level Service Request
NMVCE	7	rw	Network Management Vector Changed Service Request Enable Read: 0 _B Network Management Vector Changed Service Request disabled 1 _B Network Management Vector Changed Service Request enabled Write: 0 _B Unchanged 1 _B Disable Network Management Vector Changed Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TI0E	8	rw	Timer Service Request 0 Enable Read: 0 _B Timer Service Request 0 disabled 1 _B Timer Service Request 0 enabled Write: 0 _B Unchanged 1 _B Disable Service Request 0
TI1E	9	rw	Timer Service Request 1 Enable Read: 0 _B Timer Service Request 1 disabled 1 _B Timer Service Request 1 enabled Write: 0 _B Unchanged 1 _B Disable Timer Service Request 1
TIBCE	10	rw	Transfer Input Buffer Completed Service Request Enable Read: 0 _B Wakeup Status Service Request disabled 1 _B Wakeup Status Service Request enabled Write: 0 _B Unchanged 1 _B Disable Wakeup Status Service Request
TOBCE	11	rw	Transfer Output Buffer Completed Service Request Enable Read: 0 _B Transfer Input Buffer Completed Service Request disabled 1 _B Transfer Input Buffer Completed Service Request enabled Write: 0 _B Unchanged 1 _B Disable Transfer Input Buffer Completed Service Request
SWEE	12	rw	Stop Watch Event Service Request Enable Read: 0 _B Stop Watch Event Service Request disabled 1 Stop Watch Event Service Request enabled Write: 0 _B Unchanged 1 Disable Stop Watch Event Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SUCSE	13	rw	Startup Completed Successfully Service Request Enable Read: 0 _B Startup Completed Successfully Service Request disabled 1 _B Startup Completed Successfully Service Request enabled Write: 0 _B Unchanged 1 _B Disable Startup Completed Successfully Service Request
MBSIE	14	rw	Message Buffer Status Service Request Enable Read: 0 _B Message Buffer Status Service Request disabled 1 _B Message Buffer Status Service Request enabled Write: 0 _B Unchanged 1 _B Disable Message Buffer Status Service Request
SDSE	15	rw	Start of Dynamic Segment Service Request Enable Read: 0 _B Start of Dynamic Service Request disabled 1 _B Start of Dynamic Service Request enabled Write: 0 _B Unchanged 1 _B Disable Start of Dynamic Service Request
WUPAE	16	rw	Wakeup Pattern Channel A Service Request Enable Read: 0 _B Wakeup Pattern Channel A Service Request disabled 1 _B Wakeup Pattern Channel A Service Request enabled Write: 0 _B Unchanged 1 _B Disable Wakeup Pattern Channel A Service Request

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
MTSAE	17	rw	Media Access Test Symbol Channel A Service Request Enable Read: 0 _B Media Access Test Symbol Channel A Service Request disabled 1 _B Media Access Test Symbol Channel A Service Request enabled Write: 0 _B Unchanged 1 _B Disable Media Access Test Symbol Channel A Service Request
WUPBE	24	rw	Wakeup Pattern Channel B Service Request Enable Read: 0 _B Wakeup Pattern Channel B Service Request disabled 1 _B Wakeup Pattern Channel B Service Request enabled Write: 0 _B Unchanged 1 _B Disable Wakeup Pattern Channel A Service Request
MTSBE	25	rw	Media Access Test Symbol Channel B Service Request Enable Read: 0 _B Media Access Test Symbol Channel B Service Request disabled 1 _B Media Access Test Symbol Channel B Service Request enabled Write: 0 _B Unchanged 1 _B Disable Media Access Test Symbol Channel B Service Request
0	[23:18], [31:26]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Service Request Line Enable (ILE)

Each of the two service request lines to the Host (eray_int0, eray_int1) can be enabled / disabled separately by programming bit EINT0 and EINT1.

ILE

Service Request Line Enable (0040_H) Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														EINT 1	EINT 0
r														rw	rw

Field	Bits	Type	Description
EINT0	0	rw	Enable Service Request Line 0 0 _B Service Request line eray_int0 disabled 1 _B Service Request line eray_int0 enabled
EINT1	1	rw	Enable Service Request Line 1 0 _B Service Request line eray_int1 disabled 1 _B Service Request line eray_int1 enabled
0	[31:2]	r	Reserved Returns 0 if read; should be written with 0.

Timer 0 Configuration (T0C)

Absolute timer. Specifies in terms of cycle count and macrotick the point in time when the timer 0 service request occurs. When the timer 0 service request is asserted, output signal eray_tint0 is set to 1 for the duration of one macrotick and **SIR.TIO** is set to 1.

Timer 0 can be activated as long as the POC is either in "NORMAL_ACTIVE" state or in "NORMAL_PASSIVE" state. Timer 0 is deactivated when leaving "NORMAL_ACTIVE" state or "NORMAL_PASSIVE" state except for transitions between the two states.

Before reconfiguration of the timer, the timer has to be halted first by writing 0 to bit T0RC.

FlexRay Protocol Controller (E-Ray)

T0C

Timer 0 Configuration

(0044_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		TOMO													
r		rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	TOCC							0					TOM S	TOR C	
r		rw							r					rw	rw

Field	Bits	Type	Description
T0RC	0	rwh	Timer 0 Run Control 0 _B Timer 0 halted 1 _B Timer 0 running
T0MS	1	rw	Timer 0 Mode Select 0 _B Single-shot mode 1 _B Continuous mode
TOCC	[14:8]	rw	Timer 0 Cycle Code The 7-bit timer 0 cycle code determines the cycle set used for generation of the timer 0 service request. For details about the configuration of the cycle code see “Cycle Counter Filtering” on Page 2-203 .
TOMO	[29:16]	rw	Timer 0 Macrotick Offset Configures the macrotick offset from the beginning of the cycle where the service request is to occur. The Timer 0 Service Request occurs at this offset for each cycle of the cycle set.
0	[7:2], 15, [31:30]	r	Reserved Returns 0 if read; should be written with 0.

Note: The configuration of timer 0 is compared against the macrotick counter value, there is no separate counter for timer 0. In case the Communication Controller leaves “NORMAL_ACTIVE” or “NORMAL_PASSIVE” state, or if timer 0 is halted by Host command, output signal eray_tint0 is reset to 0 immediately.

FlexRay Protocol Controller (E-Ray)

Timer 1 Configuration (T1C)

Relative timer. After the specified number of macroticks has expired, the timer 1 service request is asserted, output signal `eray_tint1` is set to 1 for the duration of one macrotick and **SIR.T11** is set to 1.

Timer 1 can be activated as long as the POC is either in "NORMAL_ACTIVE" state or in "NORMAL_PASSIVE" state. Timer 1 is deactivated when leaving "NORMAL_ACTIVE" state or "NORMAL_PASSIVE" state except for transitions between the two states.

Before reconfiguration of the timer, the timer has to be halted first by resetting bit T1RC to 0.

T1C

Timer 1 Configuration (0048_H) Reset Value: 0002 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		T1MC													
r		rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0													T1M	T1R	
r													rw	rw	h

Field	Bits	Type	Description
T1RC	0	rw	Timer 1 Run Control 0 _B Timer 1 halted 1 _B Timer 1 running
T1MS	1	rw	Timer 1 Mode Select 0 _B Single-shot mode 1 _B Continuous mode
T1MC	[29:16]	rw	Timer 1 Macrotick Count When the configured macrotick count is reached the timer 1 service request is generated. Valid values are: 2 _H ... 3FFF _H macroticks in continuous mode 1 _H ... 3FFF _H macroticks in single-shot mode
0	[15:2], [31:30]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Note: In case the Communication Controller leaves "NORMAL_ACTIVE" or "NORMAL_PASSIVE" state, or if timer 1 is halted by Host command, output signal eray_tint1 is reset to 0 immediately.

Stop Watch Register 1 (STPW1)

The stop watch is activated by a rising or falling edge on pin eray_stpwt, by a service request 0 or 1 event (rising edge on signal eray_int0 or eray_int 1) or by the Host by writing bit **STPW1.SSWT** to 1. With the macrotick counter increment following next to the stop watch activation the actual cycle counter and macrotick value are captured in the Stop Watch Register 1 **STPW1** while the slot counter values for channel A and B are captured in the Stop Watch Register 2 **STPW2**.

STPW1

Stop Watch Register 1

(004C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		SMTV													
r		rh													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		SCCV						0	EINT 1	EINT 0	EET P	SSW T	EDG E	SWM S	ESW T
r		rh						r	rw	rw	rw	rwh	rw	rw	rwh

Field	Bits	Type	Description
ESWT	0	rwh	Enable Stop Watch Trigger If enabled an edge on input eray_stpwt or an service request 0 or 1 event (rising edge on signal or) activates the stop watch. In single-shot mode this bit is reset to 0 after the actual cycle counter and macrotick value are stored in the Stop Watch register. 0 _B Stop watch trigger disabled 1 _B Stop watch trigger enabled
SWMS	1	rw	Stop Watch Mode Select 0 _B Single-shot mode 1 _B Continuous mode
EDGE	2	rw	Stop Watch Trigger Edge Select 0 _B Falling Edge 1 _B Rising Edge

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SSWT	3	rwh	Software Stop Watch Trigger When the Host writes this bit to 1 the stop watch is activated. After the actual cycle counter and macrotick value are stored in the Stop Watch register this bit is reset to 0. The bit is only writeable while ESWT = 0. 0 _B Software trigger reset 1 _B Stop watch activated by software trigger
EETP	4	rw	Enable External Trigger Pin Enables stop watch trigger event via pin eray_stpwt if ESWT = 1. 0 _B Stop watch trigger via pin eray_stpwt disabled 1 _B Edge on pin eray_stpwt triggers stop watch
EINT0	5	rw	Enable Service Request 0 Trigger Enables stop watch trigger by service request 0 event if ESWT = 1. 0 _B Stop watch trigger by service request 0 disabled 1 _B Service Request 0 event triggers stop watch
EINT1	6	rw	Enable Service Request 1 Trigger Enables stop watch trigger by service request 1 event if ESWT = 1. 0 _B Stop watch trigger by service request 1 disabled 1 _B Service Request 1 event triggers stop watch
SCCV	[13:8]	rh	Stopped Cycle Counter Value State of the cycle counter when the stop watch event occurred. Valid values are: 0...3F _H Valid Values
SMTV	[29:16]	rh	Stopped Macrotick Value State of the macrotick counter when the stop watch event occurred. Valid values are: 0...3F _H Valid Values
0	7, [15:14], [31:30]	r	Reserved Returns 0 if read; should be written with 0.

*Note: Bits **ESWT** and **SSWT** cannot be set to 1 simultaneously. In this case the write access is ignored, and both bits keep their previous values. Therefore either the external stop watch trigger or the software stop watch trigger may be used.*

FlexRay Protocol Controller (E-Ray)

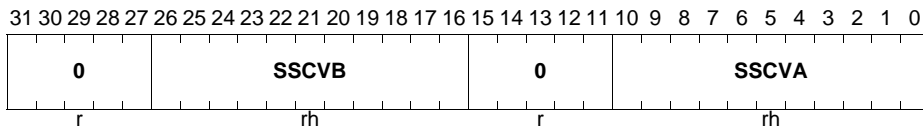
Stop Watch Register 2 (STPW2)

STPW2

Stop Watch Register 2

(0050_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
SSCVA	[10:0]	rh	Stop Watch Captured Slot Counter Value Channel A State of the slot counter for channel A when the stop watch event occurred. Valid values are 0 to 2047 (0 _H to 7FF _H).
SSCVB	[26:16]	rh	Stop Watch Captured Slot Counter Value Channel B State of the slot counter for channel B when the stop watch event occurred. Valid values are 0 to 2047 (0 _H to 7FF _H).
0	[15:11], [31:27]	r	Reserved Returns 0 if read; should be written with 0.

2.4.2.4 Communication Controller Control Registers

This section describes the registers provided by the Communication Controller to allow the Host to control the operation of the Communication Controller. The FlexRay protocol specification requires the Host to write application configuration data in “CONFIG” state only. Please consider that the configuration registers are not locked for writing in “DEFAULT_CONFIG” state.

The configuration data is reset when “DEFAULT_CONFIG” state is entered from hard reset. To change POC state from “DEFAULT_CONFIG” to “CONFIG” state the Host has to apply CHI command “CONFIG”. If the Host wants the Communication Controller to leave “CONFIG” state, the Host has to proceed as described on [“Lock Register \(LCK\)” on Page 2-26](#).

FlexRay Protocol Controller (E-Ray)

SUC Configuration Register 1 (SUCC1)

SUCC1

SUC Configuration Register 1

(0080_H)

Reset Value: 0C40 1080_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				CCH B	CCH A	MTS B	MTS A	HCS E	TSM	WUC S	PTA				
r				rw	rw	rw	rw	rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSA				0	TXS Y	TXS T	P BSY	0			CMD				
rw				r	rw	rw	rh	r			rwh				

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
CMD	[3:0]	rw	<p>CHI Command Vector</p> <p>The host may write any CHI command at any time, but certain commands are only enabled in specific POC states. A disabled command will not be executed, the CHI command vector CMD will be reset to 0000_B = "COMMAND_NOT_ACCEPTED", and flag EIR.CNA in the Error Service Request register will be set to 1. In case the previous CHI command has not yet completed, EIR.CCL is set to 1 together with EIR.CNA; the CHI command needs to be repeated. Except for HALT state, POC state change command applied while the Communication Controller is already in the requested POC state will be ignored.</p> <p>0000_B COMMAND_NOT_ACCEPTED"</p> <p>0001_B CONFIG</p> <p>0010_B READY</p> <p>0011_B WAKEUP</p> <p>0100_B RUN</p> <p>0101_B ALL_SLOTS</p> <p>0110_B HALT</p> <p>0111_B FREEZE</p> <p>1000_B SEND_MTS</p> <p>1001_B ALLOW_COLDSTART</p> <p>1010_B RESET_STATUS_INDICATORS</p> <p>1011_B MONITOR_MODE</p> <p>1100_B CLEAR_RAMs</p> <p>1101_B Reserved</p> <p>1110_B Reserved</p> <p>1111_B Reserved</p> <p>Reading SUCC1.CMD shows whether the last CHI command was accepted. CCSV.POCS monitors the actual POC state. The reserved CHI commands code hardware test functions.</p>
PBSY	7	rh	<p>POC Busy</p> <p>Signals that the POC is busy and cannot accept a command from the Host. SUCC1.CMD is locked against write accesses. Set to 1 after hard reset during initialization of internal RAM blocks.</p> <p>0_B POC not busy, SUCC1.CMD writeable</p> <p>1_B POC is busy, SUCC1.CMD locked</p>

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TXST	8	rw	Transmit Startup Frame in Key Slot^{1) 2)} (pKeySlotUsedForStartup) Defines whether the key slot is used to transmit startup frames. The bit can be modified in "DEFAULT_CONFIG" or "CONFIG" state only. 0 _B No startup frame transmission in key slot, node is non-coldstarter 1 _B Key slot used to transmit startup frame, node is leading or following coldstarter
TXSY	9	rw	Transmit Sync Frame in Key Slot^{1) 2)} (pKeySlotUsedForSync) Defines whether the key slot is used to transmit sync frames. The bit can be modified in "DEFAULT_CONFIG" or "CONFIG" state only. 0 _B No sync frame transmission in key slot, node is neither sync nor coldstart node 1 _B Key slot used to transmit sync frames, node is sync node
CSA	[15:11]	rw	Cold Start Attempts¹⁾ (gColdStartAttempts) Configures the maximum number of attempts that a cold starting node is permitted to try to start up the network without receiving any valid response from another node. It can be modified in "DEFAULT_CONFIG" or "CONFIG" state only. Must be identical in all nodes of a cluster. Valid values are 2 to 31.
PTA	[20:16]	rw	Passive to Active¹⁾ (pAllowPassiveToActive) Defines the number of consecutive even / odd cycle pairs that must have valid clock correction terms before the Communication Controller is allowed to transit from "NORMAL_PASSIVE" to "NORMAL_ACTIVE" state. If set to 00000 _B the Communication Controller is not allowed to transit from "NORMAL_PASSIVE" to "NORMAL_ACTIVE" state. It can be modified in "DEFAULT_CONFIG" or "CONFIG" state only. Valid values are 0 to 31 even / odd cycle pairs.
WUCS	21	rw	Wakeup Channel Select¹⁾ (pWakeupChannel) With this bit the Host selects the channel on which the Communication Controller sends the Wakeup pattern. The Communication Controller ignores any attempt to change the status of this bit when not in "DEFAULT_CONFIG" or "CONFIG" state. 0 _B Send wakeup pattern on channel A 1 _B Send wakeup pattern on channel B

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TSM	22	rw	<p>Transmission Slot Mode¹⁾ (pSingleSlotEnabled) Selects the initial transmission slot mode. In SINGLE slot mode the Communication Controller may only transmit in the preconfigured key slot. The key slot ID is configured in the header section of message buffer 0 respectively message buffers 0 and 1 depending on bit MRC.SPLM. In case SUCC1.TSM = 1, message buffer 0 respectively message buffers 0,1 can be (re)configured in "DEFAULT_CONFIG" or "CONFIG" state only. In ALL slot mode the Communication Controller may transmit in all slots. The bit can be written in "DEFAULT_CONFIG" or "CONFIG" state only. The communication controller changes to ALL slot mode when the Host successfully applied the ALL_SLOTS command by writing SUCC1.CMD = 0101_B in POC states "NORMAL_ACTIVE" or "NORMAL_PASSIVE". The actual slot mode is monitored by CCSV.SLM.</p> <p>0_B ALL Slot Mode 1_B SINGLE Slot Mode (default after hard reset)</p>
HCSE	23	rw	<p>Halt due to Clock Sync Error¹⁾ (pAllowHaltDueToClock) Controls the transition to "HALT" state due to a clock synchronization error. The bit can be modified in "DEFAULT_CONFIG" or "CONFIG" state only.</p> <p>0_B Communication Controller will enter / remain in "NORMAL_PASSIVE" 1_B Communication Controller will enter "HALT" state</p>
MTSA	24	rw	<p>Select Channel A for MTS Transmission^{1) 3)} The bit selects channel A for MTS symbol transmission. The flag is reset by default and may be modified only in "DEFAULT_CONFIG" or "CONFIG" state.</p> <p>0_B Channel A disabled for MTS transmission 1_B Channel A selected for MTS transmission</p>
MTSB	25	rw	<p>Select Channel B for MTS Transmission^{1) 3)} The bit selects channel B for MTS symbol transmission. The flag is reset by default and may be modified only in "DEFAULT_CONFIG" or "CONFIG" state.</p> <p>0_B Channel B disabled for MTS transmission 1_B Channel B selected for MTS transmission</p>

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
CCHA	26	rw	Connected to Channel A¹⁾ (pChannels) Configures whether the node is connected to channel A. 0 _B Not connected to channel A 1 _B Node connected to channel A (default after hard reset)
CCHB	27	rw	Connected to Channel B¹⁾ (pChannels) Configures whether the node is connected to channel B. 0 _B Not connected to channel B 1 _B Node connected to channel B (default after hard reset)
0	[6:4], 10, [31:28]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

2) The protocol requires that both bits TXST and TXSY are set for coldstart nodes.

3) **SUCC1** If both bits MTSA and MTSB are set to 1 an MTS symbol will be transmitted on both channels when requested by writing **SUCC1.CMD** = 1000_B.

COMMAND_NOT_ACCEPTED

SUCC1.CMD is reset to 0000_B due to one of the following conditions:

- Illegal command applied by the Host
- Host writes command_not_accepted
- Host applied new command while execution of the previous Host command has not completed

When **SUCC1.CMD** is reset to 0000_B, bit **EIR.CNA** in the Error Service Request register is set, and - if enabled - an service request is generated. Commands which are not accepted are not executed.

CONFIG

Go to POC state "CONFIG" when called in POC states "DEFAULT_CONFIG", "READY", or in "MONITOR_MODE". When called in "HALT" state transits to POC state "DEFAULT_CONFIG". When called in any other state, **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED".

READY

Go to POC state "READY" when called in POC states "CONFIG", "NORMAL_ACTIVE", "NORMAL_PASSIVE", "STARTUP", or "WAKEUP". When called in any other state, **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED".

WAKEUP

Go to POC state WAKEUP when called in POC state "READY". When called in any other state, **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED".

RUN

Go to POC state "STARTUP" when called in POC state "READY". When called in any other state, **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED".

ALL_SLOTS

Leave SINGLE slot mode after successful startup / integration at the next end of cycle when called in POC states "NORMAL_ACTIVE" or "NORMAL_PASSIVE". When called in any other state, **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED".

HALT

Set the halt request **CCSV.HRQ** bit in the Communication Controller Status Vector register and go to POC state "HALT" at the next end of cycle when called in POC states "NORMAL_ACTIVE" or "NORMAL_PASSIVE". When called in any other state, **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED".

FREEZE

Set the freeze status indicator **CCSV.FSI** and go to POC state "HALT" immediately. Can be called from any state.

SEND_MTS

Send single MTS symbol during the next following symbol window on the channel configured by **SUCC1.MTSA**, **SUCC1.MTSB**, when called in POC state "NORMAL_ACTIVE". When called in any other state, **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED".

ALLOW_COLDSTART

The command resets bit **CCSV.CSI** to enable the node to become cold starter. When called in states "DEFAULT_CONFIG", "CONFIG", "HALT", or "MONITOR_MODE". **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED". To become leading coldstarter it is also required that both TXST and TXSY are set.

RESET_STATUS_INDICATORS

Resets status flags **CCSV.FSI**, **CCSV.HRQ**, **CCSV.CSNI**, **CCSV.CSAI**, **CCSV.WSV** and Register **CCEV** to their default values. Flags internally evaluated in the actual POC state are not reset. May be called in any state.

MONITOR_MODE

Enter MONITOR_MODE when called in POC state CONFIG. In this mode the Communication Controller is able to receive FlexRay frames and wakeup pattern. It is also able to detect coding errors. The temporal integrity of received frames is not checked. This mode can be used for debugging purposes, e.g. in case that the startup of a FlexRay network fails. When called in any other state, **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED". For details see **"MONITOR_MODE"** on [Page 2-188](#).

CLEAR_RAMs

Sets bit **MHDS.CRAM** in the Message Handler Status register when called in "DEFAULT_CONFIG" or "CONFIG" state. When called in any other state, **SUCC1.CMD** will be reset to 0000_B = "COMMAND_NOT_ACCEPTED". **MHDS.CRAM** is also set when the Communication Controller leaves hard reset. By setting **MHDS.CRAM** all internal RAM blocks are initialized to zero. During the initialization of the RAMs, **SUCC1.PBSY** will show POC busy. Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAMs.

The initialization of the E-Ray internal RAM blocks requires 2048 f_{CLC_ERAY} cycles. There should be no Host access to IBF or OBF during initialization of the internal RAM blocks after hard reset or after assertion of CHI command CLEAR_RAMs. Before asserting CHI command CLEAR_RAMs the Host should make sure that no transfer between Message RAM and IBF / OBF or the Transient Buffer RAMs is ongoing. This command also resets the Message Buffer Status registers **MHDS**, **LDTS**, **FSR**, **MHDF**, **TXRQ1**, **TXRQ2**, **TXRQ3**, **TXRQ4**, **NDAT1**, **NDAT2**, **NDAT3**, **NDAT4**, **MBSC1**, **MBSC2**, **MBSC3**, and **MBSC4**.

Note: All accepted commands with exception of CLEAR_RAMs and SEND_MTS will cause a change of register CCSV after at most 8 cycles of the slower of the two clocks eray_bclk and eray_sclk, counted from the falling edge of the CHI input signal eray_select, assumed that POC was not busy when the command was applied and that no POC state change was forced by bus activity in that time frame. Reading register CCSV will show data that is delayed by synchronization from eray_sclk to eray_bclk domain and by the Host-specific CPU interface.

Table 2-4 below references the CHI commands from the FlexRay Protocol Specification v2.1 (section 2.2.1.1, Table 2-2) to the E-Ray CHI command vector CMD.]

Table 2-4 Reference to CHI Host command summary from FlexRay protocol specification

CHI Command	Where processed (POC State)	CHI Command Vector CMD
ALL_SLOT	POC:normal active, POC:normal passive	ALL_SLOTS
ALLOW_COLDSTART	All except POC:default config, POC:config, POC:halt	ALLOW_COLDSTART
CONFIG	POC:default config, POC:ready	CONFIG
CONFIG_COMPLETE	POC:config	Unlock sequence & READY
DEFAULT_CONFIG	POC:halt	CONFIG
FREEZE	All	FREEZE
HALT	POC:normal active, POC:normal passive	HALT
READY	All except POC:default config, POC:config, POC:ready, POC:halt	READY
RUN	POC:ready	RUN
WAKEUP	POC:ready	WAKEUP

FlexRay Protocol Controller (E-Ray)

SUC Configuration Register 2 (SUCC2)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

SUCC2

SUC Configuration Register 2 (0084_H) Reset Value: 0100 0504_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				LTN				0				LT																			
r				rw				r				rw																			

Field	Bits	Type	Description
LT	[20:0]	rw	Listen Timeout¹⁾ (pdListenTimeout) Configures wakeup / startup listen timeout in microticks. The range for wakeup / startup listen timeout (pdListenTimeout) is 1284 to 1283846 (504 _H to 1F580 _H) microticks
LTN	[27:24]	rw	Listen Timeout Noise¹⁾ (gListenNoise - 1) Configures the upper limit for startup and wakeup listen timeout in the presence of noise expressed as a multiple of the cluster constant pdListenTimeout. The range of pdListenTimeout 2 to 16. LTN must be configured identical in all nodes of a cluster. 1 _H Listen Timeout Noise is equal 2 2 _H Listen Timeout Noise is equal 3 ...B ... F _H Listen Timeout Noise is equal 16
0	[23:21], [31:28]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

Note: The wakeup / startup noise timeout is calculated as follows:

$$\begin{aligned} \text{The wakeup / startup noise timeout} &= \text{pdListenTimeout} \cdot \text{gListenNoise} \\ &= \text{LT} \cdot (\text{LTN} + 1) \end{aligned}$$

FlexRay Protocol Controller (E-Ray)

SUC Configuration Register 3 (SUCC3)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

SUCC3

SUC Configuration Register 3

(0088_H)

Reset Value: 0000 0011_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0																							WCF			WCP					
r																							rw			rw					

Field	Bits	Type	Description
WCP	[3:0]	rw	Maximum Without Clock Correction Passive¹⁾ (gMaxWithoutClockCorrectionPassive) Defines the number of consecutive even / odd cycle pairs with missing clock correction terms that will cause a transition from "NORMAL_ACTIVE" to "NORMAL_PASSIVE" state. Must be identical in all nodes of a cluster. Valid values are 1 to 15 (1 _H to F _H) cycle pairs.
WCF	[7:4]	rw	Maximum Without Clock Correction Fatal¹⁾ (gMaxWithoutClockCorrectionFatal) Defines the number of consecutive even / odd cycle pairs with missing clock correction terms that will cause a transition from "NORMAL_ACTIVE" or "NORMAL_PASSIVE" to "HALT" state. Must be identical in all nodes of a cluster. Valid values are 1 to 15 (1 _H to F _H) cycle pairs.
0	[31:8]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

NEM Configuration Register (NEMC)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

NEMC

NEM Configuration Register (008C_H) **Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0																														NML	
r																														rw	

Field	Bits	Type	Description
NML	[3:0]	rw	Network Management Vector Length¹⁾ (gNetworkManagementVectorLength) These bits configure the length of the NM vector. The configured length must be identical in all nodes of a cluster. Valid values are 0 to 12 (0 _H to C _H) bytes.
0	[31:4]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

PRT Configuration Register 1 (PRTC1)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

PRTC1

PRT Configuration Register 1 (0090_H) **Reset Value: 084C 0633_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWP				0	RXW										BRP		SPP		0	1	CASM					TSST					
rw				r	r										rw		rw		r	r	rw					rw					

Field	Bits	Type	Description
TSST	[3:0]	rw	Transmission Start Sequence Transmitter¹⁾ (gdTSSTtransmitter) Configures the duration of the Transmission Start Sequence (TSS) in terms of bit times (1 bit time = 4 microticks = 100ns at 10Mbps). Must be identical in all nodes of a cluster. Valid values are 3 to 15 (3 _H to F _H) bit times.
CASM	[10:4]	rw	Collision Avoidance Symbol Maximum¹⁾ (gdCASRxLowMax) Configures the upper limit of the acceptance window for a collision avoidance symbol (CAS). Valid values are 67 to 99 (43 _H to 63 _H). Most significant bit of CASM is hard wired to 1 and can not be modified.
SPP	[13:12]	rw	Strobe Point Position¹⁾ Defines the sample count value for strobing. The strobed bit value is set to the voted value when the sample count is incremented to the value configured by SPP. 00 _B Sample 5 (default) 01 _B Sample 4 10 _B Sample 6 11 _B Reserved; should not be used. <i>Note: The current revision 2.1 of the FlexRay protocol requires that SPP = 00_B. The alternate strobe point positions could be used to compensate for asymmetries in the physical layer.</i>

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
BRP	[15:14]	rw	<p>Baud Rate Prescaler¹⁾ (gdSampleClockPeriod, pSamplePerMicrotick)</p> <p>The baud rate prescaler configures the baud rate on the FlexRay bus. The baud rates listed below are valid with a sample clock $f_{SCLK} = 80$ MHz. One bit time always consists of 8 samples independent of the configured baud rate.</p> <p>00_B 10 MBit/s (1 microtick = 25 ns; twice sampled with f_{SCLK}) gdSampleClockPeriod = 12.5 ns = $1 / f_{SCLK}$ pSamplesPerMicrotick = 2</p> <p>01_B 5 MBit/s (1 microtick = 25ns; single sampled with $f_{SCLK} / 2$) gdSampleClockPeriod = 25 ns = $2 / f_{SCLK}$ pSamplesPerMicrotick = 1</p> <p>10_B 2.5 MBit/s (1 microtick = 50ns; single sampled with $f_{SCLK} / 4$) gdSampleClockPeriod = 50 ns = $4 / f_{SCLK}$ pSamplesPerMicrotick = 1</p> <p>11_B Reserved; should not be used (2.5 MBit/s (1 microtick = 50 ns; single sampled with $f_{SCLK} / 4$) gdSampleClockPeriod = 50 ns = $4 / f_{SCLK}$ pSamplesPerMicrotick = 1</p>
RXW	[24:16]	rw	<p>Wakeup Symbol Receive Window Length¹⁾ (gdWakeupSymbolRxWindow)</p> <p>Configures the number of bit times used by the node to test the duration of the received wakeup pattern. Must be identical in all nodes of a cluster. Valid values are 76 to 301 ($4C_H$ to $12D_H$) bit times.</p>
RWP	[31:26]	rw	<p>Repetitions of Tx Wakeup Pattern¹⁾ (pWakeupPattern)</p> <p>Configures the number of repetitions (sequences) of the Tx wakeup symbol. Valid values are 2 to 63 (2_H to $3F_H$).</p>
0	11, 25	r	<p>Reserved</p> <p>Returns 0 if read; should be written with 0.</p>

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

PRT Configuration Register 2 (PRTC2)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

PRTC2

PRT Configuration Register 2

(0094_H)

Reset Value: 0F2D 0A0E_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		TXL								TXI				0		RXL				0		RXI									
r		rw								rw				r		rw				r		rw									

Field	Bits	Type	Description
RXI	[5:0]	rw	Wakeup Symbol Receive Idle¹⁾ (gdWakeupSymbolRxIdle) Configures the number of bit times used by the node to test the duration of the idle phase of the received wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 14 to 59 (E _H to 3B _H) bit times.
RXL	[13:8]	rw	Wakeup Symbol Receive Low¹⁾ (gdWakeupSymbolRxLow) Configures the number of bit times used by the node to test the duration of the low phase of the received wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 10 to 55 (A _H to 37 _H) bit times.
TXI	[23:16]	rw	Wakeup Symbol Transmit Idle¹⁾ (gdWakeupSymbolTxIdle) Configures the number of bit times used by the node to transmit the idle phase of the wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 45 to 180 (2D _H to B4 _H) bit times.
TXL	[29:24]	rw	Wakeup Symbol Transmit Low¹⁾ (gdWakeupSymbolTxLow) Configures the number of bit times used by the node to transmit the low phase of the wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 15 to 60 (F _H to 3C _H) bit times.
0	[7:6], [15:14], [31:30]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

MHD Configuration Register (MHDC)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

MHDC

MHD Configuration Register

(0098_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		SLT														0						SFDL									
r		rw														r						rw									

Field	Bits	Type	Description
SFDL	[6:0]	rw	Static Frame Data Length (gPayloadLengthStatic) ¹⁾ Configures the cluster-wide payload length for all frames sent in the static segment in double byte. The payload length must be identical in all nodes of a cluster. Valid values are 0 to 127 (0 to 7F _H).
SLT	[28:16]	rw	Start of Latest Transmit (pLatestTx) ¹⁾ Configures the maximum minislot value allowed before inhibiting frame transmission in the dynamic segment of the cycle. There is no transmission dynamic segment if SLT is reset to zero. Valid values are 0 to 7981 (0 to 1F2D _H) minislots.
0	[15:7], [31:29]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

GTU Configuration Register 1 (GTUC01)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

GTUC01

GTU Configuration Register 1

(00A0_H)

Reset Value: 0000 0280_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												UT																			
r												rw																			

Field	Bits	Type	Description
UT	[19:0]	rw	Microtick per Cycle (pMicroPerCycle) ¹⁾ Configures the duration of the communication cycle in microticks. Valid values are 640 to 640000 (280 _H to 9C400 _H) microticks.
0	[31:20]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

GTU Configuration Register 2 (GTUC02)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

GTUC02

GTU Configuration Register 2 **(00A4_H)** **Reset Value: 0002 000A_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												SNM				0		MPC													
r												rw				r		rw													

Field	Bits	Type	Description
MPC	[13:0]	rw	MacroTICK Per Cycle (gMacroPerCycle) ¹⁾ Configures the duration of one communication cycle in macroticks. The cycle length must be identical in all nodes of a cluster. Valid values are 10 to 16000 (A _H to 3E80 _H) macroticks.
SNM	[19:16]	rw	Sync Node Max (gSyncNodeMax) ¹⁾ Maximum number of frames within a cluster with sync frame indicator bit SYN set to 1. Must be identical in all nodes of a cluster. Valid values are 2 to 15 (2 _H to F _H).
0	[15:14], [31:20]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

GTU Configuration Register 3 (GTUC03)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

GTUC03

GTU Configuration Register 3

(00A8_H)

Reset Value: 0202 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				MIOB				0				MIOA				UIOB								UIOA							
r				rw				r				rw				rw								rw							

Field	Bits	Type	Description
UIOA	[7:0]	rw	Microtick Initial Offset Channel A¹⁾ (pMicroInitialOffset[A]) Configures the number of microticks between the actual time reference point on channel A and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on pDelayCompensation[A] and therefore has to be set for each channel independently. Valid values are 0 to 240 (0 _H to F0 _H) microticks.
UIOB	[15:8]	rw	Microtick Initial Offset Channel B¹⁾ (pMicroInitialOffset[B]) Configures the number of microticks between the actual time reference point on channel B and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on pDelayCompensation[B] and therefore has to be set for each channel independently. Valid values are 0 to 240 (0 _H to F0 _H) microticks.
MIOA	[22:16]	rw	Macrotick Initial Offset Channel A (gMacroInitialOffset[A]) ¹⁾ Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration. Must be identical in all nodes of a cluster. Valid values are 2 to 72 (2 _H to 48 _H) macroticks.
MIOB	[30:24]	rw	Macrotick Initial Offset Channel B (gMacroInitialOffset[B]) ¹⁾ Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration. Must be identical in all nodes of a cluster. Valid values are 2 to 72 (2 _H to 48 _H) macroticks.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
0	23, 31	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

.GTU Configuration Register 4 (GTUC04)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only. For details about configuration of **NIT** and **OCS** see "**Configuration of Network Idle Time (NIT) Start and Offset Correction Start**" on Page 2-177.

GTUC04

GTU Configuration Register 4 (00AC_H) **Reset Value: 0008 0007_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		OCS														0		NIT													
r		rw														r		rw													

Field	Bits	Type	Description
NIT	[13:0]	rw	Network Idle Time Start ¹⁾ (gMacroPerCycle - gdNIT - 1) Configures the starting point of the Network Idle Time (NIT) at the end of the communication cycle expressed in terms of macroticks from the beginning of the cycle. The start of network idle time (NIT) is recognized if MacroTICK = gMacroPerCycle - gdNIT - 1 and the increment pulse of MacroTICK is set. Must be identical in all nodes of a cluster. Valid values are 7 to 15997 (7 _H to 3E7D _H) macroticks.
OCS	[29:16]	rw	Offset Correction Start ¹⁾ (gOffsetCorrectionStart - 1) Determines the start of the offset correction within the network idle time (NIT) phase, calculated from start of cycle. Must be identical in all nodes of a cluster. For cluster consisting of E-Ray implementations only, it is sufficient to program OCS = NIT + 1. Valid values are 8 to 15998 (8 _H to 3E7E _H) macroticks.
0	[15:14], [31:30]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

GTU Configuration Register 5 (GTUC05)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

GTUC05

GTU Configuration Register 5 (00B0_H) Reset Value: 0E00 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEC								0		CDD				DCB								DCA									
rw								r		rw				rw								rw									

Field	Bits	Type	Description
DCA	[7:0]	rw	Delay Compensation Channel A¹⁾ (pDelayCompensation[A]) Used to compensate for reception delays on channel A. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125μs to 0.05μs. In practice, the minimum of the propagation delays of all sync nodes should be applied. Valid values are 0 to 200 (0 _H to C8 _H) microticks.
DCB	[15:8]	rw	Delay Compensation Channel B¹⁾ (pDelayCompensation[B]) Used to compensate for reception delays on channel B. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05μs. In practice, the minimum of the propagation delays of all sync nodes should be applied. Valid values are 0 to 200 (0 _H to C8 _H) microticks.
CDD	[20:16]	rw	Cluster Drift Damping (pClusterDriftDamping) ¹⁾ Configures the cluster drift damping value used in clock synchronization to minimize accumulation of rounding errors. Valid values are 0 to 20 (0 _H to 14 _H) microticks.
DEC	[31:24]	rw	Decoding Correction (pDecodingCorrection) ¹⁾ Configures the decoding correction value used to determine the primary time reference point. Valid values are 14 to 143 (E _H to 8F _H) microticks.
0	[23:21]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

GTU Configuration Register 6 (GTUC06)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

GTUC06

GTU Configuration Register 6 (00B4_H) **Reset Value: 0002 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		MOD														0		ASR													
r		rw														r		rw													

Field	Bits	Type	Description
ASR	[10:0]	rw	Accepted Startup Range¹⁾ (pdAcceptedStartupRange) Number of microticks constituting the expanded range of measured deviation for startup frames during integration. Valid values are 0 to 1875 (0 _H to 753 _H) microticks.
MOD	[26:16]	rw	Maximum Oscillator Drift (pdMaxDrift) ¹⁾ Maximum drift offset between two nodes that operate with unsynchronized clocks over one communication cycle in microticks. Valid values are 2 to 1923 (2 _H to 783 _H) microticks.
0	[15:11], [31:27]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

GTU Configuration Register 7 (GTUC07)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

GTUC07

GTU Configuration Register 7 (00B8_H) **Reset Value: 0002 0004_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				NSS												0				SSL											
r				rw												r				rw											

Field	Bits	Type	Description
SSL	[9:0]	rw	Static Slot Length (gdStaticSlot) ¹⁾ Configures the duration of a static slot in macroticks. The static slot length must be identical in all nodes of a cluster. Valid values are 4 to 659 (4 _H to 293 _H) macroticks.
NSS	[25:16]	rw	Number of Static Slots (gNumberOfStaticSlots) ¹⁾ Configures the number of static slots in a cycle. At least 2 coldstart nodes must be configured to startup a FlexRay network. The number of static slots must be identical in all nodes of a cluster. Valid values are 2 to 1023 (2 _H to 3FF _H).
0	[15:10], [31:26]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

GTU Configuration Register 8 (GTUC08)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

GTUC08

GTU Configuration Register 8 (00BC_H) **Reset Value: 0000 0002_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		NMS														0										MSL					
r		rw														r										rw					

Field	Bits	Type	Description
MSL	[5:0]	rw	Minislot Length (gdMinislot) ¹⁾ Configures the duration of a minislot in macroticks. The minislot length must be identical in all nodes of a cluster. Valid values are 2 to 63 (2 _H to 3F _H) macroticks.
NMS	[28:16]	rw	Number of Minislots (gNumberOfMinislots) ¹⁾ Configures the number of minislots within the dynamic segment of a cycle. The number of minislots must be identical in all nodes of a cluster. Valid values are 0 to 7986 (0 _H to 1F32 _H).
0	[15:6], [31:29]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

GTU Configuration Register 9 (GTUC09)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

GTUC09

GTU Configuration Register 9 (00C0_H) **Reset Value: 0000 0101_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														DSI	0	MAPO				0	APO										
r														rw	r	rw				r	rw										

Field	Bits	Type	Description
APO	[5:0]	rw	Action Point Offset (gdActionPointOffset) ¹⁾ Configures the action point offset in macroticks within static slots and symbol window. Must be identical in all nodes of a cluster. Valid values are 1 to 63 (1 _H to 3F _H) macroticks.
MAPO	[12:8]	rw	Minislot Action Point Offset ¹⁾ (gdMinislotActionPointOffset) Configures the action point offset in macroticks within the minislots of the dynamic segment. Must be identical in all nodes of a cluster. Valid values are 1 to 31 (1 _H to 1F _H) macroticks.
DSI	[17:16]	rw	Dynamic Slot Idle Phase ¹⁾ (gdDynamicSlotIdlePhase) The duration of the dynamic slot idle phase has to be greater or equal than the idle detection time. Must be identical in all nodes of a cluster. Valid values are 0 to 2 Minislot.
0	[7:6], [15:13], [31:18]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

GTU Configuration Register 10 (GTUC10)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

GTUC10

GTU Configuration Register 10 (00C4_H) **Reset Value: 0002 0005_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				MRC												0				MOC											
r				rw												r				rw											

Field	Bits	Type	Description
MOC	[13:0]	rw	Maximum Offset Correction¹⁾ (pOffsetCorrectionOut) Holds the maximum permitted offset correction value to be applied by the internal clock synchronization algorithm (absolute value). The Communication Controller checks only the internal offset correction value against the maximum offset correction value. Valid values are 5 to 15266 (5 _H to 3BA2 _H) microticks.
MRC	[26:16]	rw	Maximum Rate Correction¹⁾ (pRateCorrectionOut) Holds the maximum permitted rate correction value to be applied by the internal clock synchronization algorithm. The communication controller checks only the internal rate correction value against the maximum rate correction value (absolute value). Valid values are 2 to 1923 (2 _H to 783 _H) microticks.
0	[15:14], [31:27]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

GTU Configuration Register 11 (GTUC11)

GTUC11

GTU Configuration Register 11 (00C8_H) Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				ERC				0				EOC				0				ERC C				0				EOC C			
r				rw				r				rw				r				rw				rw				rw			

Field	Bits	Type	Description
EOCC	[1:0]	rw	External Offset Correction Control (pExternOffsetControl) By writing to EOCC the external offset correction is enabled as specified below. Should be modified only outside network idle time (NIT). 00 _B No external clock correction 01 _B No external clock correction 10 _B External offset correction value subtracted from calculated offset correction value 11 _B External offset correction value added to calculated offset correction value
ERCC	[9:8]	rw	External Rate Correction Control (pExternRateControl) By writing to ERCC the external rate correction is enabled as specified below. Should be modified only outside network idle time (NIT). 00 _B No external rate correction 01 _B No external rate correction 10 _B External rate correction value subtracted from calculated rate correction value 11 _B External rate correction value added to calculated rate correction value
EOC	[18:16]	rw	External Offset Correction¹⁾ (pExternOffsetCorrection) Holds the external clock offset correction value in microticks to be applied by the internal synchronization algorithm. The value is subtracted / added from / to the calculated offset correction value. The value is applied during network idle time (NIT). May be modified in "DEFAULT_CONFIG" or "CONFIG" state only. Valid values are 0 to 7 microticks.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
ERC	[26:24]	rw	External Rate Correction¹⁾ (pExternRateCorrection) Holds the external rate correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted / added from / to the calculated rate correction value. The value is applied during network idle time (NIT). May be modified in "DEFAULT_CONFIG" or "CONFIG" state only. Valid values are 0 to 7 microticks.
0	[7:2], [15:10], [23:19], [31:27]	r	Reserved Returns 0 if read; should be written with 0.

1) This bit can be updated in "DEFAULT_CONFIG" or "CONFIG" state only!

FlexRay Protocol Controller (E-Ray)

2.4.2.5 Communication Controller Status Registers

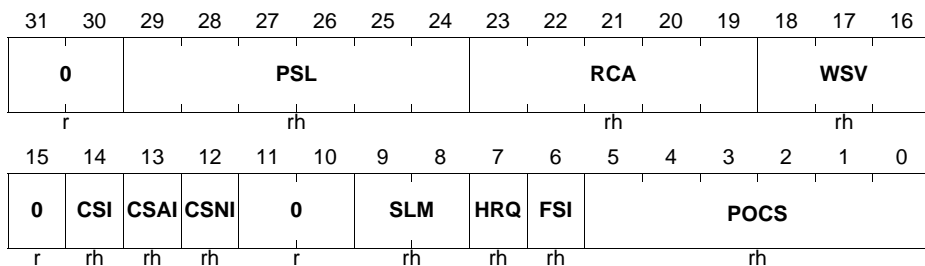
During 8/16-bit accesses to status variables coded with more than 8/16-bit, the variable might be updated by the Communication Controller between two accesses (non-atomic read accesses). The status vector may change faster than the Host can poll the status vector, depending on eray_bclk frequency.

Communication Controller Status Vector (CCSV)

CCSV

Communication Controller Status Vector(0100_H)

Reset Value: 0010 4000_H



FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
POCS	[5:0]	rh	<p>Protocol Operation Control Status</p> <p>Indicates the actual state of operation of the Communication Controller Protocol Operation Control</p> <p>000000_B "DEFAULT_CONFIG" state</p> <p>000001_B "READY" state</p> <p>000010_B "NORMAL_ACTIVE" state</p> <p>000011_B "NORMAL_PASSIVE" state</p> <p>000100_B "HALT" state</p> <p>000101_B "MONITOR_MODE" state</p> <p>000101_B ... 001110_B are reserved.</p> <p>001111_B "CONFIG" state</p> <p>Indicates the actual state of operation of the POC in the wakeup path</p> <p>010000_B WAKEUP_STANDBY state</p> <p>010001_B "WAKEUP_LISTEN" state</p> <p>010010_B "WAKEUP_SEND" state</p> <p>010011_B "WAKEUP_DETECT" state</p> <p>010100_B ... 011111_B are reserved.</p> <p>Indicates the actual state of operation of the POC in the startup path</p> <p>100000_B "STARTUP_PREPARE" state</p> <p>100001_B "COLDSTART_LISTEN" state</p> <p>100010_B "COLDSTART_COLLISION_RESOLUTION" state</p> <p>100011_B "COLDSTART_CONSISTENCY_CHECK" state</p> <p>100100_B "COLDSTART_GAP" state</p> <p>100101_B "COLDSTART_JOIN" State</p> <p>100110_B "INTEGRATION_COLDSTART_CHECK" state</p> <p>100111_B "INTEGRATION_LISTEN" state</p> <p>101000_B "INTEGRATION_CONSISTENCY_CHECK" state</p> <p>101001_B "INITIALIZE_SCHEDULE" state</p> <p>101010_B "ABORT_STARTUP" state</p> <p>101011_B "STARTUP_SUCCESS" state</p> <p>101100_B ... 111111_B are reserved.</p>

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
FSI	6	rh	Freeze Status Indicator (vPOC!Freeze) Indicates that the POC has entered the "HALT" state due to CHI command "FREEZE" or due to an error condition requiring an immediate POC halt. Reset by CHI command "RESET_STATUS_INDICATORS" or by transition from "HALT" to "DEFAULT_CONFIG" state.
HRQ	7	rh	Halt Request (vPOC!CHI!HaltRequest) Indicates that a request from the Host has been received to halt the POC at the end of the communication cycle. Reset by CHI command "RESET_STATUS_INDICATORS", by transition from "HALT" to "DEFAULT_CONFIG" state or when entering "READY" state.
SLM	[9:8]	rh	Slot Mode (vPOC!SlotMode) Indicates the actual slot mode of the POC. Default is "SINGLE". Changes to "ALL", depending on configuration bit SUCC1.TSM . In "NORMAL_ACTIVE" or "NORMAL_PASSIVE" state the CHI command "ALL_SLOTS" will change the slot mode from "SINGLE" over "ALL_PENDING" to "ALL". When not in "NORMAL_ACTIVE" or "NORMAL_PASSIVE" state then reset by CHI command "RESET_STATUS_INDICATORS" to the value defined by SUCC1.TSM . 00 _B SINGLE 01 _B Reserved 10 _B ALL_PENDING 11 _B ALL
CSNI	12	rh	Coldstart Noise Indicator (vPOC!ColdstartNoise) Indicates that the cold start procedure occurred under noisy conditions. Reset by CHI command "RESET_STATUS_INDICATORS" or by transition from "HALT" to "DEFAULT_CONFIG" state or from "READY" to "STARTUP" state.
CSAI	13	rh	Coldstart Abort Indicator Coldstart aborted. Reset by CHI command "RESET_STATUS_INDICATORS" or by transition from "HALT" to "DEFAULT_CONFIG" state or from "READY" to "STARTUP" state.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
CSI	14	rh	<p>Cold Start Inhibit (vColdStartInhibit)</p> <p>Indicates that the node is disabled from cold starting. The flag is set whenever the POC enters "READY" state due to CHI command "READY". The flag has to be reset under control of the Host by CHI command "ALLOW_COLDSTART" (SUCC1.CMD = 1001_B).</p> <p>0_B Cold starting of node enabled</p> <p>1_B Cold starting of node disabled</p>
WSV	[18:16]	rh	<p>Wakeup Status (vPOC!WakeupStatus)</p> <p>Indicates the status of the current wakeup attempt. Reset by CHI command "RESET_STATUS_INDICATORS" or by transition from "HALT" to "DEFAULT_CONFIG" state or from "READY" to "STARTUP" state.</p> <p>000_B UNDEFINED. Wakeup not yet executed by the Communication Controller.</p> <p>001_B RECEIVED_HEADER. Set when the Communication Controller finishes wakeup due to the reception of a frame header without coding violation on either channel in "WAKEUP_LISTEN" state.</p> <p>010_B RECEIVED_WUP. Set when the Communication Controller finishes wakeup due to the reception of a valid wakeup pattern on the configured wakeup channel in "WAKEUP_LISTEN" state.</p> <p>011_B COLLISION_HEADER. Set when the Communication Controller stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid header on either channel.</p> <p>100_B COLLISION_WUP. Set when the Communication Controller stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid wakeup pattern on the configured wakeup channel.</p> <p>101_B COLLISION_UNKNOWN. Set when the Communication Controller stops wakeup by leaving "WAKEUP_DETECT" state after expiration of the wakeup timer without receiving a valid wakeup pattern or a valid frame header.</p> <p>110_B TRANSMITTED. Set when the Communication Controller has successfully completed the transmission of the wakeup pattern.</p> <p>111_B Reserved</p>

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
RCA	[23:19]	rh	Remaining Coldstart Attempts (vRemainingColdstartAttempts) Indicates the number of remaining coldstart attempts. The RUN command resets this counter to the maximum number of coldstart attempts as configured by SUCC1.CSA .
PSL	[29:24]	rh	POC Status Log Status of CCSV.POCS immediately before entering "HALT" state. Set when entering "HALT" state. Set to "HALT" when FREEZE command is applied during "HALT" state. Reset to 000000 _B when leaving "HALT" state.
0	[11:10], 15, [31:30]	rh	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

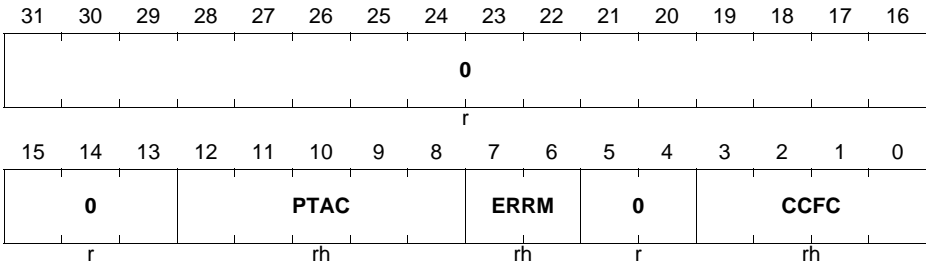
Communication Controller Error Vector (CCEV)

Reset by CHI command RESET_STATUS_INDICATORS or by transition from “HALT” to “DEFAULT_CONFIG” state or when entering “READY” state.

CCEV

Communication Controller Error Vector (0104_H)

Reset Value: 0000 0000_H



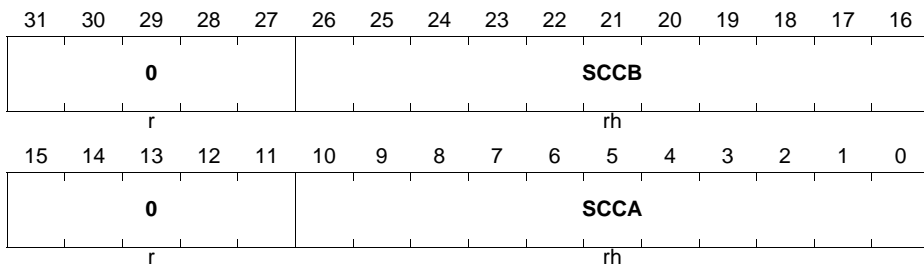
Field	Bits	Type	Description
CCFC	[3:0]	rh	Clock Correction Failed Counter (vClockCorrectionFailed) The Clock Correction Failed Counter is incremented by one at the end of any odd communication cycle where either the missing offset correction error or missing rate correction error are active. The Clock Correction Failed Counter is reset to 0 at the end of an odd communication cycle if neither the offset correction failed nor the rate correction failed errors are active. The Clock Correction Failed Counter stops at 15.
ERRM	[7:6]	rh	Error Mode (vPOC!ErrorMode) Indicates the actual error mode of the POC. 00 _B “ACTIVE” (green) 01 _B “PASSIVE” (yellow) 10 _B “COMM_HALT” (red) 11 _B Reserved
PTAC	[12:8]	rh	Passive to Active Count (vAllowPassiveToActive) Indicates the number of consecutive even / odd cycle pairs that have passed with valid rate and offset correction terms, while the node is waiting to transit from “NORMAL_PASSIVE” state to “NORMAL_ACTIVE” state. The transition takes place when PTAC equals SUCC1.PTA .
0	[5:4], [31:13]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Slot Counter Value (SCV)

SCV

Slot Counter Value (0110_H) **Reset Value: 0000 0000_H**



Field	Bits	Type	Description
SCCA	[10:0]	rh	Slot Counter Channel A (vSlotCounter[A]) Current slot counter value on channel A. The value is incremented by the Communication Controller and reset at the start of a communication cycle. Valid values are 0 to 2047 (0 _H to 7FD _H).
SCCB	[26:16]	rh	Slot Counter Channel B (vSlotCounter[B]) Current slot counter value on channel B. The value is incremented by the Communication Controller and reset at the start of a communication cycle. Valid values are 0 to 2047 (0 _H to 7FD _H).
0	[15:11], [31:27]	r	Reserved Returns 0 if read; should be written with 0.

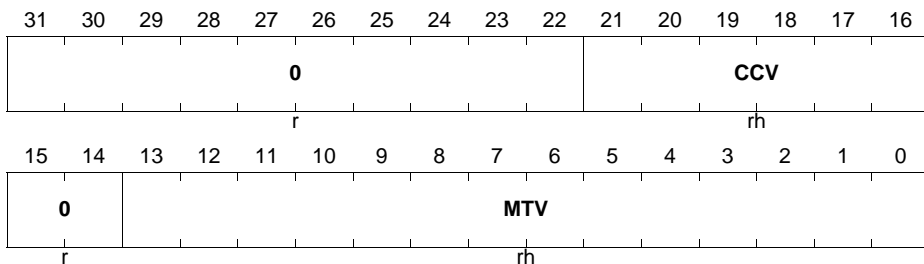
FlexRay Protocol Controller (E-Ray)

Macrotick and Cycle Counter Value (MTCCV)

MTCCV

Macrotick and Cycle Counter Value (0114_H)

Reset Value: 0000 0000_H



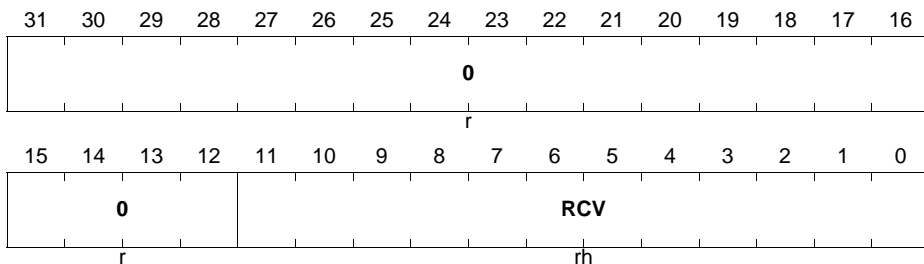
Field	Bits	Type	Description
MTV	[13:0]	rh	Macrotick Value (vMacrotick) Current macrotick value. The value is incremented by the Communication Controller and reset at the start of a communication cycle. Valid values are 0 to 16000 (0 _H to 3E80 _H).
CCV	[21:16]	rh	Cycle Counter Value (vCycleCounter) Current cycle counter value. The value is incremented by the Communication Controller at the start of a communication cycle. Valid values are 0 to 63 (0 _H to 3F _H).
0	[15:14], [31:22]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Rate Correction Value (RCV)

RCV

Rate Correction Value (0118_H) **Reset Value: 0000 0000_H**



Field	Bits	Type	Description
RCV	[11:0]	rh	Rate Correction Value (vRateCorrection) Rate correction value (two's complement). Calculated internal rate correction value before limitation. If the RCV value exceeds the limits defined by GTUC10.MRC , flag SFS.RCLR is set to 1.
0	[31:12]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

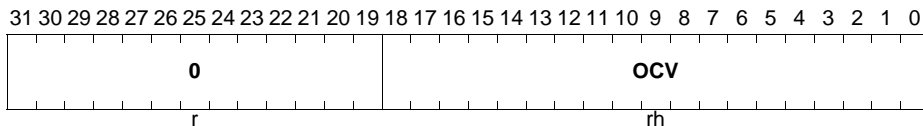
Offset Correction Value (OCV)

OCV

Offset Correction Value

(011C_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
OCV	[18:0]	rh	Offset Correction Value (vOffsetCorrection) Offset correction value (two's complement). Calculated internal offset correction value before limitation. If the OCV value exceeds the limits defined by GTUC10.MOC flag SFS.OCLR is set to 1.
0	[31:19]	r	Reserved Returns 0 if read; should be written with 0.

Note: The external rate / offset correction value is added to the limited rate / offset correction value.

FlexRay Protocol Controller (E-Ray)

Sync Frame Status (SFS)

The maximum number of valid sync frames in a communication cycle is 15.

SFS

Sync Frame Status (0120_H) **Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0												RC LR	MR CS	OC LR	MO CS
r												rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VSBO				VSBE				VSAO				VSAE			
rh				rh				rh				rh			

Field	Bits	Type	Description
VSAE	[3:0]	rh	Valid Sync Frames Channel A, even communication cycle Holds the number of valid sync frames received on channel A in the even communication cycle. If transmission of sync frames is enabled by SUCC1.TXSY the value is incremented by one. The value is updated during the network idle time (NIT) of each even communication cycle. This bit field is only valid if the channel A is assigned to the Communication Controller by SUCC1.CCHA .
VSAO	[7:4]	rh	Valid Sync Frames Channel A, odd communication cycle Holds the number of valid sync frames received on channel A in the odd communication cycle. If transmission of sync frames is enabled by SUCC1.TXSY the value is incremented by one. The value is updated during the network idle time (NIT) of each odd communication cycle. This bit field is only valid if the channel A is assigned to the Communication Controller by SUCC1.CCHA .
VSBE	[11:8]	rh	Valid Sync Frames Channel B, even communication cycle Holds the number of valid sync frames received on channel B in the even communication cycle. If transmission of sync frames is enabled by SUCC1.TXSY the value is incremented by one. The value is updated during the network idle time (NIT) of each even communication cycle. This bit field is only valid if the channel B is assigned to the Communication Controller by SUCC1.CCHB .

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
VSBO	[15:12]	rh	Valid Sync Frames Channel B, odd communication cycle Holds the number of valid sync frames received on channel B in the odd communication cycle. If transmission of sync frames is enabled by SUCC1.TXSY the value is incremented by one. The value is updated during the network idle time (NIT) of each odd communication cycle. This bit field is only valid if the channel B is assigned to the Communication Controller by SUCC1.CCHB .
MOCS	16	rh	Missing Offset Correction Signal The Missing Offset Correction flag signals to the Host, that no offset correction calculation can be performed because no sync frames were received. The flag is updated by the Communication Controller at start of offset correction phase. 0 _B Offset correction signal valid 1 _B Missing offset correction signal
OCLR	17	rh	Offset Correction Limit Reached The Offset Correction Limit Reached flag signals to the Host, that the offset correction value has exceeded its limit as defined by GTUC10.MOC . The flag is updated by the Communication Controller at start of offset correction phase. 0 _B Offset correction below limit 1 _B Offset correction limit reached
MRCS	18	rh	Missing Rate Correction Signal The Missing Rate Correction Flag signals to the Host, that no rate correction calculation can be performed because no pairs of even / odd sync frames were received. The flag is updated by the Communication Controller at start of offset correction phase. 0 _B Rate correction signal valid 1 _B Missing rate correction signal
RCLR	19	rh	Rate Correction Limit Reached The Rate Correction Limit Reached flag signals to the Host, that the rate correction value has exceeded its limit.as defined by GTUC10.MRC . The flag is updated by the Communication Controller at start of offset correction phase. 0 _B Rate correction below limit 1 _B Rate correction limit reached
0	[31:20]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Symbol Window and network idle time (NIT) Status (SWNIT)

Symbol window related status information. Updated by the Communication Controller at the end of the symbol window for each channel. During startup the status data is not updated.

*Note: MTSA and MTSB may be changed outside “DEFAULT_CONFIG” or “CONFIG” state when the write to **SUC Configuration Register 1 (SUCC1)** register is directly preceded by the unlock sequence as described in “**Lock Register (LCK)**” on Page 2-26. This may be combined with CHI command SEND_MTS. If both bits MTSA and MTSB are set to 1 an MTS symbol will be transmitted on both channels when requested by writing **SUCC1.CMD** = 1000_B*

SWNIT

Symbol Window and Network Idle Time Status

(0124_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				SBN	SEN	SBN	SEN	MTS	MTS	TCS	SBS	SES	TCS	SBS	SES
r				B	B	A	A	B	A	B	B	B	A	A	A
				rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
SESA	0	rh	Syntax Error in Symbol Window Channel A (vSSISyntaxErrorA) 0 _B No syntax error detected 1 _B Syntax error during symbol window detected on channel A
SBSA	1	rh	Slot Boundary Violation in Symbol Window Channel A (vSSIBViolationA) 0 _B No slot boundary violation detected 1 _B Slot boundary violation during symbol window detected on channel A

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TCSA	2	rh	Transmission Conflict in Symbol Window Channel A (vSSI!TxConflictA) 0 _B No transmission conflict detected 1 _B Transmission conflict in symbol window detected on channel A
SESB	3	rh	Syntax Error in Symbol Window Channel B (vSSI!SyntaxErrorB) 0 _B No syntax error detected 1 _B Syntax error during symbol window detected on channel B
SBSB	4	rh	Slot Boundary Violation in Symbol Window Channel B (vSSI!BViolationB) 0 _B No slot boundary violation detected 1 _B Slot boundary violation during symbol window detected on channel B
TCSB	5	rh	Transmission Conflict in Symbol Window Channel B (vSSI!TxConflictB) 0 _B No transmission conflict detected 1 _B Transmission conflict in symbol window detected on channel B
MTSA	6	rh	MTS Received on Channel A (vSSI!ValidMTSA) ¹⁾ Media Access Test symbol received on channel A during the proceeding symbol window. Updated by the Communication Controller for each channel at the end of the symbol window. When this bit is set to 1, also interrupt flag SIR.MTSA is set to 1. 0 _B No MTS symbol received on channel A 1 _B MTS symbol received on channel A
MTSB	7	rh	MTS Received on Channel B (vSSI!ValidMTSB) ¹⁾ Media Access Test symbol received on channel B during the proceeding symbol window. Updated by the Communication Controller for each channel at the end of the symbol window. When this bit is set to 1, also interrupt flag SIR.MTSB is set to 1. 0 _B No MTS symbol received on channel B 1 _B MTS symbol received on channel B

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SENA	8	rh	Syntax Error during network idle time (NIT) Channel A (vSSI!SyntaxErrorA) Updated by the Communication Controller channel A at the end of the NIT. 0 _B No syntax error detected 1 _B Syntax error during network idle time (NIT) detected on channel A
SBNA	9	rh	Slot Boundary Violation during network idle time (NIT) Channel A (vSSI!BViolationA) Updated by the Communication Controller channel A at the end of the NIT. 0 _B No slot boundary violation detected 1 _B Slot boundary violation during network idle time (NIT) detected on channel A
SENB	10	rh	Syntax Error during network idle time (NIT) Channel B (vSSI!SyntaxErrorB) Updated by the Communication Controller channel B at the end of the NIT. 0 _B No syntax error detected 1 _B Syntax error during network idle time (NIT) detected on channel B
SBNB	11	rh	Slot Boundary Violation during network idle time (NIT) Channel B (vSSI!BViolationB) Updated by the Communication Controller channel B at the end of the NIT. 0 _B No slot boundary violation detected 1 _B Slot boundary violation during network idle time (NIT) detected on channel B
0	[31:12]	r	Reserved Returns 0 if read; should be written with 0.

- 1) MTSA and MTSB may also be changed outside "DEFAULT_CONFIG" or "CONFIG" state when the write to **SUC Configuration Register 1 (SUCC1)** register is directly preceded by the unlock sequence as described in "**Lock Register (LCK)**" on **Page 2-26**. This may be combined with CHI command SEND_MTS. If both bits MTSA and MTSB are set to 1 an MTS symbol will be transmitted on both channels when requested by writing **SUCC1.CMD** = 1000_B.

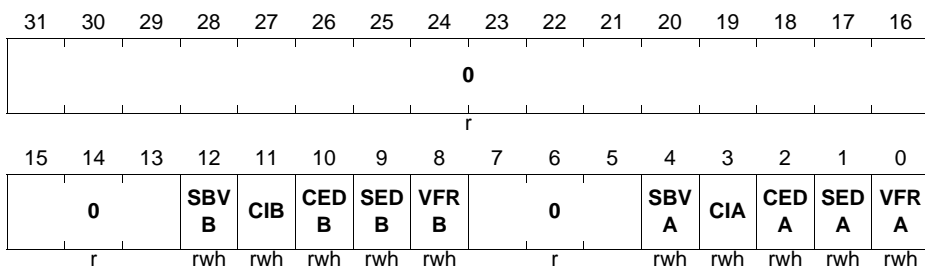
FlexRay Protocol Controller (E-Ray)

Aggregated Channel Status (ACS)

The aggregated channel status provides the Host with an accrued status of channel activity for all communication slots regardless of whether they are assigned for transmission or subscribed for reception. The aggregated channel status also includes status data from the symbol window and the network idle time. The status data is updated (set) after each slot and aggregated until it is reset by the Host. During startup the status data is not updated. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect on the flag. A hard reset will also clear the register.

ACS

Aggregated Channel Status (0128_H) **Reset Value: 0000 0000_H**



Field	Bits	Type	Description
VFRA	0	rwh	Valid Frame Received on Channel A (vSS!ValidFrameA) One or more valid frames were received on channel A in any static or dynamic slot during the observation period. 0 _B No valid frame received 1 _B Valid frame(s) received on channel A
SEDA	1	rwh	Syntax Error Detected on Channel A (vSS!SyntaxErrorA) One or more syntax errors in static or dynamic slots, symbol window, and network idle time (NIT) were observed on channel A. 0 _B No syntax error observed 1 _B Syntax error(s) observed on channel A
CEDA	2	rwh	Content Error Detected on Channel A (vSS!ContentErrorA) One or more frames with a content error were received on channel A in any static or dynamic slot during the observation period. 0 _B No frame with content error received 1 _B Frame(s) with content error received on channel A

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
CIA	3	rwh	Communication Indicator Channel A One or more valid frames were received on channel A in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation. 0_B No valid frame(s) received in slots containing any additional communication 1_B Valid frame(s) received on channel A in slots containing any additional communication
SBVA	4	rwh	Slot Boundary Violation on Channel A (vSS!BViolationA) One or more slot boundary violations were observed on channel A at any time during the observation period (static or dynamic slots, symbol window, and network idle time NIT). 0_B No slot boundary violation observed 1_B Slot boundary violation(s) observed on channel A
VFRB	8	rwh	Valid Frame Received on Channel B (vSS!ValidFrameB) One or more valid frames were received on channel B in any static or dynamic slot during the observation period. 0_B No valid frame received 1_B Valid frame(s) received on channel B
SEDB	9	rwh	Syntax Error Detected on Channel B (vSS!SyntaxErrorB) One or more syntax errors in static or dynamic slots, symbol window, and network idle time (NIT) were observed on channel B. 0_B No syntax error observed 1_B Syntax error(s) observed on channel B
CEDB	10	rwh	Content Error Detected on Channel B (vSS!ContentErrorB) One or more frames with a content error were received on channel B in any static or dynamic slot during the observation period. 0_B No frame with content error received 1_B Frame(s) with content error received on channel B

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
CIB	11	rwh	Communication Indicator Channel B One or more valid frames were received on channel B in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation. 0_B No valid frame(s) received in slots containing any additional communication 1_B Valid frame(s) received on channel B in slots containing any additional communication
SBVB	12	rwh	Slot Boundary Violation on Channel B (vSS!BViolationB) One or more slot boundary violations were observed on channel B at any time during the observation period (static or dynamic slots, symbol window, and network idle time NIT). 0_B No slot boundary violation observed 1_B Slot boundary violation(s) observed on channel B
0	[7:5], [31:13]	r	Reserved Returns 0 if read; should be written with 0.

*Note: The set condition of flags CIA and CIB is also fulfilled if there is only one single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase. When one of the flags SEDB, CEDB, CIB, SBVB changes from 0 to 1, service request flag **EIR.EDB** is set to 1. When one of the flags SEDA, CEDA, CIA, SBVA changes from 0 to 1, service request flag **EIR.EDA** is set to 1.*

Even Sync ID [01...15] (ESIDnn)

Registers ESID01 to ESID15 hold the frame IDs of the sync frames received in **even** communication cycles, sorted in ascending order, with register ESID01 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an even communication cycle, register ESID1 holds the respective sync frame ID as configured in message buffer 0 and the flags RXEA, RXEB are set. The value is updated during the network idle time (NIT) of each even communication cycle.

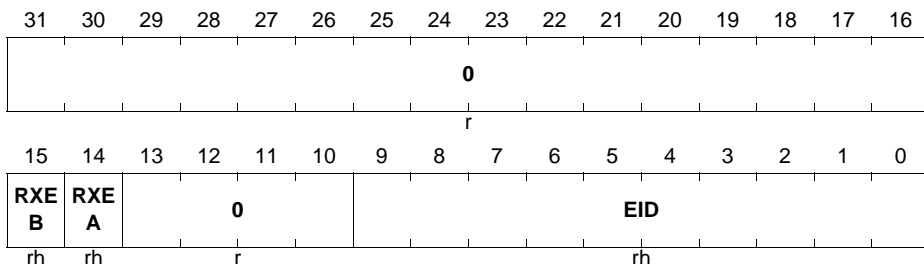
FlexRay Protocol Controller (E-Ray)

ESIDnn (nn = 01-15)

Even Sync ID Symbol Window nn

$(012C_H + nn * 4)$

Reset Value: 0000 0000_H



Field	Bits	Type	Description
EID	[9:0]	rh	Even Sync ID (vsSyncIDListA,B even) Sync frame ID even communication cycle.
RXEA	14	rh	Received/Configured Even Sync ID on Channel A Signals that a sync frame corresponding to the stored even sync ID was received on channel A or that the node is configured to be a sync node with key slot = EID (ESID1 only). 0 _B Sync frame not received on channel A / node configured to transmit sync frames 1 _B Sync frame received on channel A / node not configured to transmit sync frames
RXEB	15	rh	Received/Configured Even Sync ID on Channel B Signals that a sync frame corresponding to the stored even sync ID was received on channel B or that the node is configured to be a sync node with key slot = EID (ESID1 only). 0 _B Sync frame not received on channel B / node configured to transmit sync frames 1 _B Sync frame received on channel B / node not configured to transmit sync frames
0	[13:10], [31:16]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

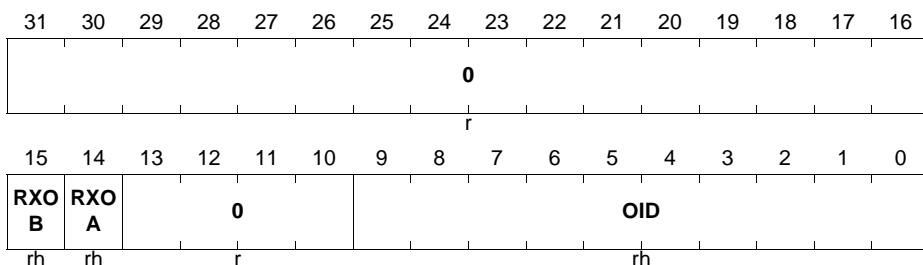
Odd Sync ID [01...15] (OSIDnn)

Registers OSID01 to OSID15 hold the frame IDs of the sync frames received in **odd** communication cycles, sorted in ascending order, with register OSID01 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an odd communication cycle, register OSID01 holds the respective sync frame ID as configured in message buffer 0 and flags RXOA, RXOB are set. The value is updated during the network idle time (NIT) of each odd communication cycle.

OSIDnn (nn = 01-15)

Odd Sync ID Symbol Window nn($016C_H + nn * 4$)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
OID	[9:0]	rh	Odd Sync ID (vsSyncIDListA,B odd) Sync frame ID even communication cycle.
RXOA	14	rh	Received Odd Sync ID on Channel A Signals that a sync frame corresponding to the stored odd sync ID was received on channel A or that the node is configured to be a sync node with key slot = OID (OSID1 only). 0 _B Sync frame not received on channel A/ node configured to transmit sync frames 1 _B Sync frame received on channel A/ node not configured to transmit sync frames
RXOB	15	rh	Received Odd Sync ID on Channel B Signals that a sync frame corresponding to the stored odd sync ID was received on channel B or that the node is configured to be a sync node with key slot = OID (OSID1 only) 0 _B Sync frame not received on channel B/ node configured to transmit sync frames 1 _B Sync frame received on channel B/ node not configured to transmit sync frames

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
0	[13:10], [31:16]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Network Management Vector [1...3] (NMVn)

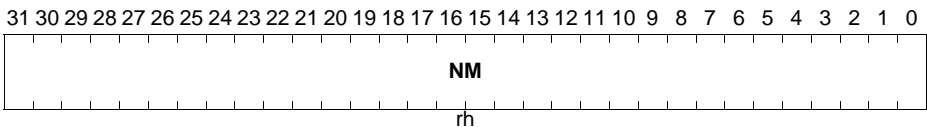
The three network management registers hold the accrued network management (NM) vector (configurable 0 to 12 byte). The accrued network management (NM) vector is generated by the Communication Controller by bit-wise ORing each network management (NM) vector received (valid static frames with PPI = 1) on each channel (see [“Network Management” on Page 2-201](#)). The Communication Controller updates the Network Management (NM) vector at the end of each communication cycle as long as the Communication Controller is either in “NORMAL_ACTIVE” or “NORMAL_PASSIVE” state. NMVn-bytes exceeding the configured network management (NM) vector length are not valid.

NMVnn (nn = 1-3)

Network Management Vector nn

$$(01AC_H + nn * 4)$$

Reset Value: 0000 0000_H



Field	Bits	Type	Description
NM	[31:0]	rh	Network Management Vector

Table 2-5 below shows the assignment of the received payload's data byte to the network management vector.

Table 2-5 Assignment of Data Byte to Network Management Vector

Bit	3	3	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0		
Word	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0										
NM1	Data3								Data2								Data1								Data0							
NM2	Data7								Data6								Data5								Data4							
NM3	Data11								Data10								Data9								Data8							

2.4.2.6 Message Buffer Control Registers

Message RAM Configuration (MRC)

The Message RAM Configuration register defines the number of message buffers assigned to the static segment, dynamic segment, and FIFO. The register can be written during "DEFAULT_CONFIG" or "CONFIG" state only.

MRC

Message RAM Configuration

(0300_H)

Reset Value: 0180 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0				S P L M	SEC	LCB										FFB										FDB									
r					rw	rw	rw										rw										rw								

Field	Bits	Type	Description
FDB	[7:0]	rw	First Dynamic Buffer May be modified in "DEFAULT_CONFIG" or "CONFIG" state only. 00 _H No group of message buffers exclusively for the static segment configured 01 _H ...7F _H Message buffers 0 to FFB-1 reserved for static segment ≥80 _H No dynamic message buffers configured
FFB	[15:8]	rw	First Buffer of FIFO May be modified in "DEFAULT_CONFIG" or "CONFIG" state only. 00 _H All message buffers assigned to the FIFO 01 _H ...7F _H Message buffers from FFB to LCB assigned to the FIFO ≥80 _H No message buffers assigned to the FIFO
LCB	[23:16]	rw	Last Configured Buffer May be modified in "DEFAULT_CONFIG" or "CONFIG" state only. 01 _H ...7F _H Number of message buffers is LCB + 1 ≥80 _H No message buffer configured

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SEC	[25:24]	rw	Secure Buffers Not evaluated when the Communication Controller is in "DEFAULT_CONFIG" or "CONFIG" state.
			00 _B Reconfiguration of message buffers enabled with numbers < FFB enabled.
			<i>Note: In nodes configured for sync frame transmission or for single slot mode operation message buffer 0 (and if SPLM = 1, also message buffer 1) Reconfiguration of all message buffers is always locked</i>
			01 _B Reconfiguration of message buffers with numbers < FDB and with numbers ≥ FFB locked and transmission of message buffers for static segment with numbers ≥ FDB disabled
			10 _B Reconfiguration of all message buffers locked
SPLM	26	rw	11 _B Reconfiguration of all message buffers locked and transmission of message buffers for static segment with numbers ≥ FDB disabled
0	[31:27]	r	Sync Frame Payload Multiplex This bit is only evaluated if the node is configured as sync node (SUCC1.TXSY = 1) or for single slot mode operation (SUCC1.TSM = 1). When this bit is set to 1 message buffers 0 and 1 are dedicated for sync frame transmission with different payload data on channel A and B. When this bit is reset to 0, sync frames are transmitted from message buffer 0 with the same payload data on both channels. Note that the channel filter configuration for message buffer 0 resp. message buffer 1 has to be chosen accordingly.
			0 _B Only message buffer 0 locked against reconfiguration
			1 _B Both message buffers 0 and 1 are locked against reconfiguration
0	[31:27]	r	Reserved Returns 0 if read; should be written with 0.

*Note: In case the node is configured as sync node (**SUCC1.TXSY** = 1) or for single slot mode operation (**SUCC1.TSM** = 1), message buffer 0 resp. 1 is reserved for sync frames or single slot frames and have to be configured with the node-specific key slot ID. In case the node is neither configured as sync node nor for single slot operation message buffer 0 resp. 1 is treated like all other message buffers.*

Table 2-6 Usage of the three Message Buffer Pointer

Message Buffer 0	↓ Static Buffers	
Message Buffer 1		
	↓ Static + Dynamic Buffers	← FDB
...		FIFO configured: FFB > FDB
	↓ FIFO	← FFB No FIFO configured: FFB ≥ 128
Message Buffer N-1		LCB ≥ FDB , LCB ≥ FFB
Message Buffer N		← LCB

The programmer has to ensure that the configuration defined by FDB, FFB, and LCB is valid. **The Communication Controller does not check for erroneous configurations!**

Note: The maximum number of header sections is 128. This means a maximum of 128 message buffer can be configured. The maximum length of a data section is 254 byte. The length of the data section may be configured differently for each message buffer. For details see [“Message RAM” on Page 2-228](#).

In case two or more message buffers are assigned to slot 1 by use of cycle filtering, all of them must be located either in the “Static Buffers” or at the beginning of the “Static + Dynamic Buffers” section.

The payload length configured and the length of the data section need to be configured identical for all message buffers belonging to the FIFO via [WRHS2.PLC](#) and [WRHS3.DP](#). When the Communication Controller is not in “DEFAULT_CONFIG” or “CONFIG” state reconfiguration of message buffers belonging to the FIFO is locked.

FlexRay Protocol Controller (E-Ray)

FIFO Rejection Filter (FRF)

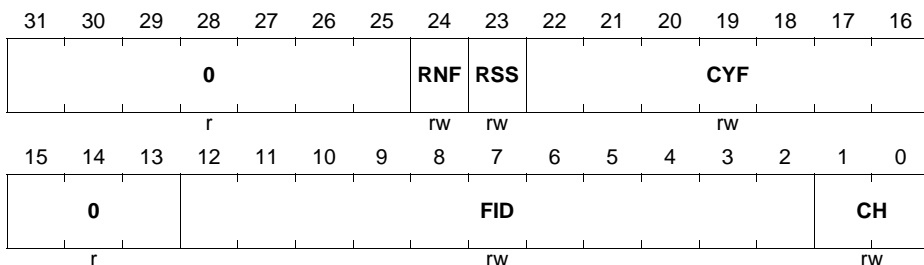
The FIFO Rejection Filter defines a user specified sequence of bits to which channel, frame ID, and cycle count of the incoming frames are compared. Together with the FIFO Rejection Filter Mask this register determines whether a message is rejected by the FIFO. The FRF register can be written during “DEFAULT_CONFIG” or “CONFIG” state only.

FRF

FIFO Rejection Filter

(0304_H)

Reset Value: 0180 0000_H



Field	Bits	Type	Description
CH	[1:0]	rw	Channel Filter May be modified in “DEFAULT_CONFIG” or “CONFIG” state only. 00 _B receive on both channels ¹⁾ 01 _B receive only on channel B 10 _B receive only on channel A 11 _B no reception
FID	[12:2]	rw	Frame ID Filter Determines the frame ID to be rejected by the FIFO. With the additional configuration of register FRFM, the corresponding frame ID filter bits are ignored, which results in further rejected frame IDs. When FRFM.MFID is zero, a frame ID filter value of zero means that no frame ID is rejected. 000 _H ... 7FF _H Frame ID filter values

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
CYF	[22:16]	rw	Cycle Counter Filter The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by CYF, all frames are rejected. For details about the configuration of the cycle counter filter see “Cycle Counter Filtering” on Page 2-203 . May be modified in “DEFAULT_CONFIG” or “CONFIG” state only.
RSS	23	rw	Reject in Static Segment If this bit is set, the FIFO is used only be used in dynamic segment. May be modified in “DEFAULT_CONFIG” or “CONFIG” state only. 0 _B FIFO also used in static segment 1 _B Reject messages for static segment
RNF	24	rw	Reject Null Frames If this bit is set, received null frames are not stored in the FIFO. May be modified in “DEFAULT_CONFIG” or “CONFIG” state only. 0 _B Null frames are stored in the FIFO 1 _B Reject all null frames
0	[15:13], [31:25]	r	Reserved Returns 0 if read; should be written with 0.

- 1) If reception on both channels is configured, also in static segment always both frames (from channel A and B) are stored in the FIFO, even if they are identical.

FlexRay Protocol Controller (E-Ray)

FIFO Rejection Filter Mask (FRFM)

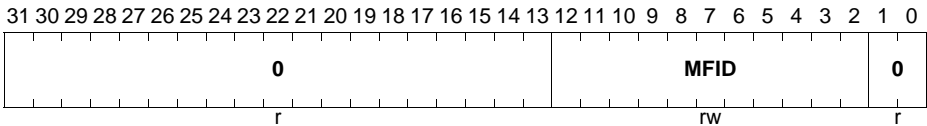
The FIFO Rejection Filter Mask specifies which of the corresponding frame ID filter bits are relevant for rejection filtering. If a bit is set, it indicates that the corresponding bit in the FRF register will not be considered for rejection filtering. The FRFM register can be written during "DEFAULT_CONFIG" or "CONFIG" state only.

FRFM

FIFO Rejection Filter Mask

(0308_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
MFID	[12:2]	rw	Mask Frame ID Filter May be modified in "DEFAULT_CONFIG" or "CONFIG" state only. 0 _B Corresponding frame ID filter bit is used for rejection filtering. 1 _B Ignore corresponding frame ID filter bit.
0	[1:0], [31:13]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

FIFO Critical Level (FCL)

The Communication Controller accepts modifications of the register in "DEFAULT_CONFIG" or "CONFIG" state only.

FCL

FIFO Critical Level (030C_H) **Reset Value: 0000 0080_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0																												CL								
r																												rw								

Field	Bits	Type	Description
CL	[7:0]	rw	Critical Level When the receive FIFO fill level FSR.RFFL is equal or greater than the critical level configured by CL, the receive FIFO critical level flag FSR.RFCL is set. If CL is programmed to values > 128, bit FSR.RFCL is never set. When FSR.RFCL changes from 0 to 1 bit SIR.RFCL is set to 1, and if enabled, an service request is generated.
0	[31:8]	r	Reserved Returns 0 if read; should be written with 0.

2.4.2.7 Message Buffer Status Registers

Message Handler Status (MHDS)

The Message Handler Status register gives the Host access to the actual state of the Message Handler. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect on the flag. A hard reset will also clear the register. When one of the flags **MHDS.PIBF**, **MHDS.POBF**, **MHDS.PMR**, **MHDS.PTBF1**, **MHDS.PTBF2** changes from 0 to 1 **EIR.PERR** is set to 1.

MHDS

Message Handler Status

(0310_H)

Reset Value: 0000 0080_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								0							
MBU								MBT							
r				rh				r				rh			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								CRA M	MFM B	FMB D	PTB F2	PTB F1	PMR	POB F	PIBF
r				rh				rh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
PIBF	0	rwh	Parity Error Input Buffer RAM 1,2 0 _B No parity error 1 _B Parity error occurred when reading Input Buffer RAM 1 or Input Buffer RAM 2
POBF	1	rwh	Parity Error Output Buffer RAM 1,2 0 _B No parity error 1 _B Parity error occurred when reading Output Buffer RAM 1 or Output Buffer RAM 2
PMR	2	rwh	Parity Error Message RAM 0 _B No parity error 1 _B Parity error occurred when reading the Message RAM
PTBF1	3	rwh	Parity Error Transient Buffer RAM A 0 _B No parity error 1 _B Parity error occurred when reading Transient Buffer RAM A

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
PTBF2	4	rwh	Parity Error Transient Buffer RAM B 0 _B No parity error 1 _B Parity error occurred when reading Transient Buffer RAM B
FMBD	5	rwh	Faulty Message Buffer Detected 0 _B No faulty message buffer 1 _B Message buffer referenced by MHDS.FMB holds faulty data due to a parity error
MFMB	6	rwh	Multiple Faulty Message Buffers detected 0 _B No additional faulty message buffer 1 _B Another faulty message buffer was detected while flag MHDS.FMBD is set
GRAM	7	rh	Clear all internal RAM's Signals that execution of the CHI command CLEAR_RAMs is ongoing (all bits of all internal RAM blocks are written to 0). The bit is set by hardware reset or by CHI command CLEAR_RAMs. 0 _B No execution of the CHI command CLEAR_RAMs 1 _B Execution of the CHI command CLEAR_RAMs ongoing
FMB	[14:8]	rh	Faulty Message Buffer Parity error occurred when reading from the message buffer or when transferring data from Input Buffer or Transient Buffer A or Transient Buffer B to the message buffer referenced by MHDS.FMB . Value only valid when one of the flags MHDS.PIBF , MHDS.PMR , MHDS.PTBF1 , MHDS.PTBF2 , and flag MHDS.FMBD is set. Updated only after the Host has reset flag MHDS.FMBD .
MBT	[22:16]	rh	Message Buffer Transmitted Number of last successfully transmitted message buffer. If the message buffer is configured for single-shot mode, the respective TXR flag in the Transmission Request Registers TXRQ1 ... TXRQ4 was reset. MBT is reset when the Communication Controller leaves "CONFIG" state or enters "STARTUP" state.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
MBU	[30:24]	rh	Message Buffer Updated Number of message buffer that was updated last. For this message buffer the respective NDn (n = 0-31) ... NDn (n = 96-127) and / or MBCn (n = 0-31) ... MBCn (n = 96-127) flag in the New Data Registers NDAT1 ... NDAT4 and the Message Buffer Status Changed MBSC1 ... MBSC4 registers are also set. MBU is reset when the Communication Controller leaves "CONFIG" state or enters "STARTUP" state.
0	15, 23, 31	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Last Dynamic Transmit Slot (LDTs)

The Last Dynamic Transmit Slot Register stores the Slot COUNTER value at the time of the last frame transmission in the dynamic segment. This register is reset when the Communication Controller leaves "CONFIG" state or enters "STARTUP" state.

LDTs

Last Dynamic Transmit Slot (0314_H) Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		LDTB														0		LDTA													
r		rh														r		rh													

Field	Bits	Type	Description
LDTA	[10:0]	rh	Last Dynamic Transmission Channel A Value of (vSlotCounter[A]) at the time of the last frame transmission on channel A in the dynamic segment of this node. It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.
LDTB	[26:16]	rh	Last Dynamic Transmission Channel B Value of (vSlotCounter[B]) at the time of the last frame transmission on channel B in the dynamic segment of this node. It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.
0	[15:11], [31:27]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

FIFO Status Register (FSR)

The register is reset when the Communication Controller leaves "CONFIG" state or enters "STARTUP" state.

FSR

FIFO Status Register (0318_H) **Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFFL								0				RFO	RFC L	RFN E	
rh								r				rh	rh	rh	

Field	Bits	Type	Description
RFNE	0	rh	Receive FIFO Not Empty This flag is set by the Communication Controller when a received valid frame (data or null frame depending on rejection mask) was stored in the FIFO. In addition, service request flag SIR.RFNE is set. The bit is reset after the Host has read all message from the FIFO. 0 Receive FIFO is empty 1 Receive FIFO is not empty
RFCL	1	rh	Receive FIFO Critical Level This flag is set when the receive FIFO fill level RFFL is equal or greater than the critical level as configured by FCL.CL . The flag is cleared by the Communication Controller as soon as RFFL drops below FCL.CL . When RFCL changes from 0 to 1 bit SIR.RFCL is set to 1, and if enabled, an service request is generated. 0 _B Receive FIFO below critical level 1 _B Receive FIFO critical level reached

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
RFO	2	rh	Receive FIFO Overflow The flag is set by the Communication Controller when a receive FIFO overflow is detected. When a receive FIFO overflow occurs, the oldest message is overwritten with the actual received message. In addition, service request flag EIR.RFO is set. The flag is cleared by the next FIFO read access issued by the Host. 0 _B No receive FIFO overflow detected 1 _B A receive FIFO overflow has been detected
RFFL	[15:8]	rh	Receive FIFO Fill Level Number of FIFO buffers filled up with new data not yet read by the Host. Maximum value is 128.
0	[7:3], [31:16]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Message Handler Constraints Flags (MHDF)

Some constraints exist for the Message Handler regarding $f_{\text{CLC_ERAY}}$ frequency, Message RAM configuration, and FlexRay bus traffic. To simplify software development, constraints violations are reported by setting flags in the MHDF. The register is reset when the Communication Controller leaves "CONFIG" state or enters "STARTUP" state.

MHDF

Message Handler Constraints Flags (031C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0							WAH P	0	WBF B	WBF A	FNF B	FNF A	SNU B	SNU A	
rh							rw	r	rw	rw	rw	rw	rw	rw	

Field	Bits	Type	Description
SNUA	0	rw	Status Not Updated Channel A This flag is set by the Communication Controller when the Message Handler, due to overload condition, was not able to update a message buffer's status MBS with respect to channel A. 0 _B No overload condition occurred when updating MBS for channel A 1 _B MBS for channel A not updated
SNUB	1	rw	Status Not Updated Channel B This flag is set by the Communication Controller when the Message Handler, due to overload condition, was not able to update a message buffer's status MBS with respect to channel B. 0 _B No overload condition occurred when updating MBS for channel B 1 _B MBS for channel B not updated

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
FNFA	2	rw	Find Sequence Not Finished Channel A This flag is set by the Communication Controller when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer) with respect to channel A. 0 _B No find sequence not finished for channel A 1 _B Find sequence not finished for channel A
FNFB	3	rw	Find Sequence Not Finished Channel B This flag is set by the Communication Controller when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer) with respect to channel B. 0 _B No find sequence not finished for channel B 1 _B Find sequence not finished for channel B
TBFA	4	rw	Transient Buffer Access Failure A This flag is set by the Communication Controller when a read or write access to Transient Buffer A requested by PRT A could not complete within the available time. 0 _B No TBF A access failure 1 _B TBF A access failure
TBFB	5	rw	Transient Buffer Access Failure B This flag is set by the Communication Controller when a read or write access to Transient Buffer B requested by PRT B could not complete within the available time. 0 _B No Transient Buffer B access failure 1 _B Transient Buffer B access failure
WAHP	8	rw	Write Attempt to Header Partition Outside "DEFAULT_CONFIG" and "CONFIG" state this flag is set by the Communication Controller when the message handler tries to write message data into the header partition of the Message RAM due to faulty configuration of a message buffer. The write attempt is not executed, to protect the header partition from unintended write accesses. 0 _B No write attempt to header partition 1 _B Write attempt to header partition
0	[7:6], [31:9]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Transmission Request 1 (TXRQ1)

This register reflect the state of the TXR flags of the configured message buffers 0 ...31. The flags are evaluated for transmit buffers only. If the number of configured message buffers is less than 31, the remaining TXRn flags have no meaning and are read as 0.

TXRQ1

Transmission Request Register 1 (0320_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXR 31	TXR 30	TXR 29	TXR 28	TXR 27	TXR 26	TXR 25	TXR 24	TXR 23	TXR 22	TXR 21	TXR 20	TXR 19	TXR 18	TXR 17	TXR 16
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXR 15	TXR 14	TXR 13	TXR 12	TXR 11	TXR 10	TXR 9	TXR 8	TXR 7	TXR 6	TXR 5	TXR 4	TXR 3	TXR 2	TXR 1	TXR 0
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
TXRn (n = 0-31)	n	rh	Transmission Request n (n = 0-31) If the flag is set, the respective message buffer 0 ... 31 is ready for transmission respectively transmission of this message buffer is in progress. In single-shot mode the flags are reset after transmission has completed.

FlexRay Protocol Controller (E-Ray)

Transmission Request Register 2 (TXRQ2)

This register reflect the state of the TXR flags of the configured message buffers 31 ... 63. The flags are evaluated for transmit buffers only. If the number of configured message buffers is less than 63, the remaining TXRn flags have no meaning and are read as 0.

TXRQ2

Transmission Request Register 2 (0324_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXR 63	TXR 62	TXR 61	TXR 60	TXR 59	TXR 58	TXR 57	TXR 56	TXR 55	TXR 54	TXR 53	TXR 52	TXR 51	TXR 50	TXR 49	TXR 48
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXR 47	TXR 46	TXR 45	TXR 44	TXR 43	TXR 42	TXR 41	TXR 40	TXR 39	TXR 38	TXR 37	TXR 36	TXR 35	TXR 34	TXR 33	TXR 32
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
TXRn (n = 32-63)	n - 32	rh	Transmission Request n (n = 32-63) If the flag is set, the respective message buffer 32 ... 63 is ready for transmission respectively transmission of this message buffer is in progress. In single-shot mode the flags are reset after transmission has completed.

FlexRay Protocol Controller (E-Ray)

Transmission Request Register 3 (TXRQ3)

This register reflect the state of the TXR flags of the configured message buffers 64 ... 95. The flags are evaluated for transmit buffers only. If the number of configured message buffers is less than 95, the remaining TXRn flags have no meaning and are read as 0.

TXRQ3

Transmission Request Register 3 (0328_H) Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXR 95	TXR 94	TXR 93	TXR 92	TXR 91	TXR 90	TXR 89	TXR 88	TXR 87	TXR 86	TXR 85	TXR 84	TXR 83	TXR 82	TXR 81	TXR 80
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXR 79	TXR 78	TXR 77	TXR 76	TXR 75	TXR 74	TXR 73	TXR 72	TXR 71	TXR 70	TXR 69	TXR 68	TXR 67	TXR 66	TXR 65	TXR 64
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
TXRn (n = 64-95)	n - 64	rh	Transmission Request n (n = 64-95) If the flag is set, the respective message buffer 64 ... 95 is ready for transmission respectively transmission of this message buffer is in progress. In single-shot mode the flags are reset after transmission has completed.

FlexRay Protocol Controller (E-Ray)

Transmission Request Register 4 (TXRQ4)

This register reflect the state of the TXR flags of the configured message buffers 96 ... 127. The flags are evaluated for transmit buffers only. If the number of configured message buffers is less than 127, the remaining TXRn flags have no meaning and are read as 0.

TXRQ4

Transmission Request Register 4 (032C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXR 127	TXR 126	TXR 125	TXR 124	TXR 123	TXR 122	TXR 121	TXR 120	TXR 119	TXR 118	TXR 117	TXR 116	TXR 115	TXR 114	TXR 113	TXR 112
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXR 111	TXR 110	TXR 109	TXR 108	TXR 107	TXR 106	TXR 105	TXR 104	TXR 103	TXR 102	TXR 101	TXR 100	TXR 99	TXR 98	TXR 97	TXR 96
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
TXRn (n = 96-127)	n - 96	rh	Transmission Request n (n = 96-127) If the flag is set, the respective message buffer 96 ... 127 is ready for transmission respectively transmission of this message buffer is in progress. In single-shot mode the flags are reset after transmission has completed.

FlexRay Protocol Controller (E-Ray)

New Data Register 1 (NDAT1)

This register reflect the state of the ND flags of all configured message buffers 0 ...31. ND flags assigned to transmit buffers are meaningless. If the number of configured message buffers is less than 31, the remaining NDn flags have no meaning. The registers are reset when the Communication Controller leaves "CONFIG" state or enters "STARTUP" state.

NDAT1

New Data Register 1

(0330_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND 31	ND 30	ND 29	ND 28	ND 27	ND 26	ND 25	ND 24	ND 23	ND 22	ND 21	ND 20	ND 19	ND 18	ND 17	ND 16
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND 15	ND 14	ND 13	ND 12	ND 11	ND 10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
NDn (n = 0-31)	n	rh	New Data n (n = 0-31) The flags are set when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer. The flags are not set after reception of null frames except for message buffers belonging to the receive FIFO. An ND flag is reset when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

FlexRay Protocol Controller (E-Ray)

New Data Register 2 (NDAT2)

This register reflect the state of the ND flags of all configured message buffers 32 ...63. ND flags assigned to transmit buffers are meaningless. If the number of configured message buffers is less than 63, the remaining NDn flags have no meaning. The registers are reset when the Communication Controller leaves "CONFIG" state or enters "STARTUP" state.

NDAT2

New Data Register 2

(0334_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND 63	ND 62	ND 61	ND 60	ND 59	ND 58	ND 57	ND 56	ND 55	ND 54	ND 53	ND 52	ND 51	ND 50	ND 49	ND 48
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND 47	ND 46	ND 45	ND 44	ND 43	ND 42	ND 41	ND 40	ND 39	ND 38	ND 37	ND 36	ND 35	ND 34	ND 33	ND 32
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
NDn (n = 32-63)	n - 32	rh	New Data n (n = 32-63) The flags are set when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer. The flags are not set after reception of null frames except for message buffers belonging to the receive FIFO. An ND flag is reset when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

FlexRay Protocol Controller (E-Ray)

New Data Register 3 (NDAT3)

This register reflect the state of the ND flags of all configured message buffers 64 ...95. ND flags assigned to transmit buffers are meaningless. If the number of configured message buffers is less than 95, the remaining NDn flags have no meaning. The registers are reset when the Communication Controller leaves "CONFIG" state or enters "STARTUP" state.

NDAT3

New Data Register 3

(0338_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND 95	ND 94	ND 93	ND 92	ND 91	ND 90	ND 89	ND 88	ND 87	ND 86	ND 85	ND 84	ND 83	ND 82	ND 81	ND 80
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND 79	ND 78	ND 77	ND 76	ND 75	ND 74	ND 73	ND 72	ND 71	ND 70	ND 69	ND 68	ND 67	ND 66	ND 65	ND 64
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
NDn (n = 64-95)	n - 64	rh	New Data n (n = 64-95) The flags are set when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer. The flags are not set after reception of null frames except for message buffers belonging to the receive FIFO. An ND flag is reset when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

FlexRay Protocol Controller (E-Ray)

New Data Register 4 (NDAT4)

This register reflect the state of the ND flags of all configured message buffers 127 ...96. ND flags assigned to transmit buffers are meaningless. If the number of configured message buffers is less than 127, the remaining NDn flags have no meaning. The registers are reset when the Communication Controller leaves "CONFIG" state or enters "STARTUP" state.

NDAT4

New Data Register 4

(033C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND 127	ND 126	ND 125	ND 124	ND 123	ND 122	ND 121	ND 120	ND 119	ND 118	ND 117	ND 116	ND 115	ND 114	ND 113	ND 112
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND 111	ND 110	ND 109	ND 108	ND 107	ND 106	ND 105	ND 104	ND 103	ND 102	ND 101	ND 100	ND 99	ND 98	ND 97	ND 96
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
NDn (n = 96-127)	n - 96	rh	New Data n (n = 96-127) The flags are set when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer. The flags are not set after reception of null frames except for message buffers belonging to the receive FIFO. An ND flag is reset when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

FlexRay Protocol Controller (E-Ray)

Message Buffer Status Changed 1 (MBSC1)

This register reflect the state of the MBC flags of all configured message buffers. If the number of configured message buffers is less than 31, the remaining MBCn flags have no meaning. The register is reset when the communication controller leaves "CONFIG" state or enters "STARTUP" state.

MBSC1

Message Buffer Status Changed 1 (0340_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBC 31	MBC 30	MBC 29	MBC 28	MBC 27	MBC 26	MBC 25	MBC 24	MBC 23	MBC 22	MBC 21	MBC 20	MBC 19	MBC 18	MBC 17	MBC 16
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MBC 15	MBC 14	MBC 13	MBC 12	MBC 11	MBC 10	MBC 9	MBC 8	MBC 7	MBC 6	MBC 5	MBC 4	MBC 3	MBC 2	MBC 1	MBC 0
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
MBCn (n = 0-31)	n	rh	Message Buffer Status Changed n (n = 0-31) An MBC flags is set whenever the Message Handler changes on of the status flags VFRA , VFRB , SEOA , SEOB , CEOA , CEOB , SVOA , SVOB , TCIA , TCIB , ESA , ESB , MLST , FTA , FTB in the header section (see " Message Buffer Status (MBS) " on Page 2-168) of the respective message buffer 0 ... 31. The flags are reset when the header section of the message buffer is reconfigured or when it has been transferred to the Output Buffer.

FlexRay Protocol Controller (E-Ray)

Message Buffer Status Changed 2 (MBSC2)

This register reflect the state of the MBC flags of all configured message buffers. If the number of configured message buffers is less than 63, the remaining MBCn flags have no meaning. The register is reset when the communication controller leaves "CONFIG" state or enters "STARTUP" state.

MBSC2

Message Buffer Status Changed 2 (0344_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBC 63	MBC 62	MBC 61	MBC 60	MBC 59	MBC 58	MBC 57	MBC 56	MBC 55	MBC 54	MBC 53	MBC 52	MBC 51	MBC 50	MBC 49	MBC 48
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MBC 47	MBC 46	MBC 45	MBC 44	MBC 43	MBC 42	MBC 41	MBC 40	MBC 39	MBC 38	MBC 37	MBC 36	MBC 35	MBC 34	MBC 33	MBC 32
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
MBCn (n = 32-63)	n - 32	rh	Message Buffer Status Changed n (n = 32-63) An MBC flags is set whenever the Message Handler changes on of the status flags VFRA , VFRB , SEOA , SEOB , CEOA , CEOB , SVOA , SVOB , TCIA , TCIB , ESA , ESB , MLST , FTA , FTB in the header section (see " Message Buffer Status (MBS) " on Page 2-168) of the respective message buffer 32 ... 63. The flags are reset when the header section of the message buffer is reconfigured or when it has been transferred to the Output Buffer.

FlexRay Protocol Controller (E-Ray)

Message Buffer Status Changed 3 (MBSC3)

This register reflect the state of the MBC flags of all configured message buffers. If the number of configured message buffers is less than 95, the remaining MBCn flags have no meaning. The register is reset when the communication controller leaves "CONFIG" state or enters "STARTUP" state.

MBSC3

Message Buffer Status Changed 3 (0348_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBC 95	MBC 94	MBC 93	MBC 92	MBC 91	MBC 90	MBC 89	MBC 88	MBC 87	MBC 86	MBC 85	MBC 84	MBC 83	MBC 82	MBC 81	MBC 80
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MBC 79	MBC 78	MBC 77	MBC 76	MBC 75	MBC 74	MBC 73	MBC 72	MBC 71	MBC 70	MBC 69	MBC 68	MBC 67	MBC 66	MBC 65	MBC 64
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
MBCn (n = 64-95)	n - 64	rh	Message Buffer Status Changed n (n = 64-95) An MBC flags is set whenever the Message Handler changes on of the status flags VFRA , VFRB , SEOA , SEOB , CEOA , CEOB , SVOA , SVOB , TCIA , TCIB , ESA , ESB , MLST , FTA , FTB in the header section (see " Message Buffer Status (MBS) " on Page 2-168) of the respective message buffer 64 ... 95. The flags are reset when the header section of the message buffer is reconfigured or when it has been transferred to the Output Buffer.

FlexRay Protocol Controller (E-Ray)

Message Buffer Status Changed 4 (MBSC4)

This register reflect the state of the MBC flags of all configured message buffers. If the number of configured message buffers is less than 127, the remaining MBCn flags have no meaning. The register is reset when the communication controller leaves "CONFIG" state or enters "STARTUP" state.

MBSC4

Message Buffer Status Changed 4 (034C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBC 127	MBC 126	MBC 125	MBC 124	MBC 123	MBC 122	MBC 121	MBC 120	MBC 119	MBC 118	MBC 117	MBC 116	MBC 115	MBC 114	MBC 113	MBC 112
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MBC 111	MBC 110	MBC 109	MBC 108	MBC 107	MBC 106	MBC 105	MBC 104	MBC 103	MBC 102	MBC 101	MBC 100	MBC 99	MBC 98	MBC 97	MBC 96
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
MBCn (n = 96-127)	n - 96	rh	Message Buffer Status Changed n (n = 96-127) An MBC flags is set whenever the Message Handler changes on of the status flags VFRA , VFRB , SEOA , SEOB , CEOA , CEOB , SVOA , SVOB , TCIA , TCIB , ESA , ESB , MLST , FTA , FTB in the header section (see "Message Buffer Status (MBS)" on Page 2-168) of the respective message buffer 96 ... 127. The flags are reset when the header section of the message buffer is reconfigured or when it has been transferred to the Output Buffer.

2.4.2.8 Identification Registers

Core Release Register (CREL)

This register contains bit-fields about the ERAY module identification. It is read only.

CREL

Core Release Register

(03F0_H)

Reset Value: XXXX XXXX_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REL				STEP				SUB STEP				YEAR				MON				DAY											
r				r				r				r				r															

Field	Bits	Type	Description
DAY	[7:0]	r	Design Time Stamp, Day Two digits, BCD-coded.
MON	[15:8]	r	Design Time Stamp, Month Two digits, BCD-coded.
YEAR	[19:16]	r	Design Time Stamp, Year One digit, BCD-coded.
SUBSTEP	[27:24]	r	Sub-Step of Core Release One digits, BCD-coded. 0 _H Alpha, pre-Beta, pre-Beta-update, pre-Beta2, pre-Beta2-update, Beta, Beta2, Revision 1.0.0 1 _H Beta_ct, Beta-ct-fix1, Revision 1.0.1 2 _H Revision1.0RC1,Beta-ct-fix2, REVISION 1.0RC1
STEP	[27:24]	r	Step of Core Release One digits, BCD-coded. 0 _H Revision 1.0.0 1 _H Alpha 2 _H pre-Beta 3 _H pre-Beta-update 4 _H pre-Beta2 5 _H pre-Beta2-update 6 _H Beta 7 _H Beta2

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
REL	[31:28]	r	Core Release One digit, BCD-coded. 0 _B alpha...beta2ct 1 _B Revision 1.0

Table 2-7 Coding of releases

Release	Step	Sub-Step	Name	Release Date
0	1	0	Alpha	
0	2	0	pre-Beta	
0	3	0	pre-Beta-update	
0	4	0	pre-Beta2	
0	5	0	pre-Beta2-update	
0	6	0	Beta	
0	6	1	Beta-ct-fix1	14.10.2005
0	6	2	Beta-ct-fix2	14.12.2005
0	7	0	Beta2	03.02.2006
0	7	1	Beta2ct	24.03.2006
0	7	2	Revision 1.0RC1	07.04.2006
1	0	0	Release 1.0.0	19.05.2006
1	0	1	Release 1.0.1	2006

Endian Register (ENDN)

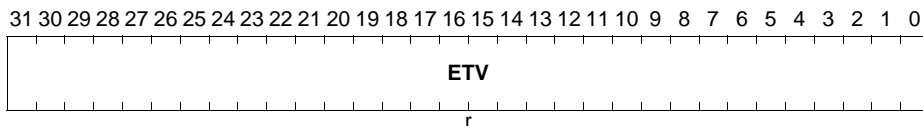
This register may be used to check, if the data of the E-Ray is handled by a host with the correct endian format. It is read only.

ENDN

Endian Register

(003F4_H)

Reset Value: 8765 4321_H



Field	Bits	Type	Description
ETV	[31:0]	r	Endianness Test Value The endianness test value.

2.4.2.9 Input Buffer

Double buffer structure consisting of Input Buffer Host and Input Buffer Shadow. While the Host can write to Input Buffer Host, the transfer to the Message RAM is done from Input Buffer Shadow. The Input Buffer holds the header and data sections to be transferred to the selected message buffer in the Message RAM. It is used to configure the message buffers in the Message RAM and to update the data sections of transmit buffers.

When updating the header section of a message buffer in the Message RAM from the Input Buffer, the Message Buffer Status as described in **“Message Buffer Status (MBS)” on Page 2-168** is automatically reset to zero.

The header sections of message buffers belonging to the receive FIFO can only be (re)configured when the Communication Controller is in “DEFAULT_CONFIG” or “CONFIG” state. For those message buffers only the payload length configured and the data pointer need to be configured via **WRHS2.PLC** and **WRHS2.DP**. All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask.

The data transfer between Input Buffer (IBF) and Message RAM is described in detail in **“Data Transfer from Input Buffer to Message RAM” on Page 2-212**.

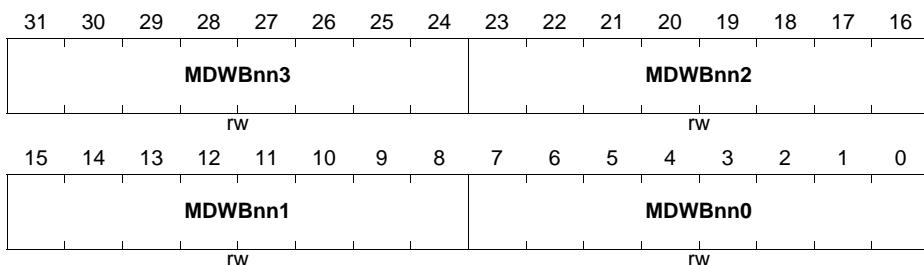
FlexRay Protocol Controller (E-Ray)

Write Data Section [1...64] (WRDSnn (nn = 01-64))

Holds the data words to be transferred to the data section of the addressed message buffer. The data words (DW_n) are written to the Message RAM in transmission order from DW1 (byte0, byte1) to DW_{PL} (PL = number of data words as defined by the payload length configured by **WRHS2.PLC**).

WRDSnn (nn = 01-64)

Write Data Section nn **(03FC_H + nn * 4)** **Reset Value: 0000 0000_H**



Field	Bits	Type	Description
MDWB0	[7:0]	rw	32-Bit Word nn, Byte 0
MDWB1	[15:8]	rw	32-Bit Word nn, Byte 1
MDWB2	[23:16]	rw	32-Bit Word nn, Byte 2
MDWB3	[31:24]	rw	32-Bit Word nn, Byte 3

Note: 16-Bit Word 127 is located on WRDS64.MDW. In this case WRDS64.MDW is unused (no valid data). The Input Buffer RAMs are initialized to zero when leaving hard reset or by CHI command CLEAR_RAMs.

1. When writing to the **WRDSnn (nn = 01-64)**, each 32-bit word has to be filled up by one 32-bit access OR two consecutive 16-bit accesses OR four consecutive 8-bit accesses before the transfer from the Input Buffer to the Message RAM is started by writing the number of the target message buffer in the Message RAM to the Input Buffer Command Request register.

Write Header Section 1 (WRHS1)

WRHS1

Write Header Section 1

(0500_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC						
r		rw	rw	rw	rw	rw	rw	r				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0					FID										
r					rw										

Field	Bits	Type	Description
FID	[10:0]	rw	Frame ID Frame ID of the selected message buffer. The frame ID defines the slot number for transmission / reception of the respective message. Message buffers with frame ID = 0 are considered as not valid.
CYC	[22:16]	rw	Cycle Code The 7-bit cycle code determines the cycle set used for cycle counter filtering. For details about the configuration of the cycle code see Section 2.5.7.3 .
CHA	24	rw	Channel Filter Control A The channel filtering field A associated with the buffer serves of channel A as a filter for receive buffers, and as a control field for transmit buffers
CHB	25	rw	Channel Filter Control B The channel filtering field B associated with the buffer serves of channel B as a filter for receive buffers, and as a control field for transmit buffers
CFG	26	rw	Message Buffer Direction Configuration Bit This bit is used to configure the corresponding buffer as a transmit buffer or as a receive buffer. For message buffers belonging to the receive FIFO the bit is not evaluated. 0 _B The corresponding buffer is configured as Receive Buffer 1 _B The corresponding buffer is configured as Transmit Buffer

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
PPIT	27	rw	Payload Preamble Indicator Transmit This bit is used to control the state of the Payload Preamble Indicator in transmit frames. If the bit is set in a static message buffer, the respective message buffer holds network management information. If the bit is set in a dynamic message buffer the first two byte of the payload segment may be used for message ID filtering by the receiver. Message ID filtering of received FlexRay frames is not supported by the E-Ray module, but can be done by the Host. 0 _B Payload Preamble Indicator not set 1 _B Payload Preamble Indicator set
TXM	28	rw	Transmission Mode This bit is used to select the transmission mode (see “Transmit Buffers” on Page 2-205). 0 _B Continuous mode 1 _B Single-shot mode
MBI	29	rw	Message Buffer Service Request This bit enables the receive / transmit service request for the corresponding message buffer. After a dedicated receive buffer has been updated by the Message Handler, flag SIR.RXI and /or SIR.MBSI in the Status Service Request register are set. After a transmission has completed flag SIR.TXI is set. 0 _B The corresponding message buffer service request is disabled 1 _B The corresponding message buffer service request is enabled
0	[15:11], 23, [31:30]	r	Reserved Returns 0 if read; should be written with 0.

Note: The Input Buffer RAMs are initialized to zero when leaving hardware reset or by CHI command CLEAR_RAMs.

Table 2-8 Channel Filter Control Bits

CHA	CHB	Transmit Buffer transmit frame on	Receive Buffer store frame received from
1 ¹⁾	1 ¹⁾	Both Channels (static segment only)	Channel A or B (store first semantically valid frame, static segment only)
1	0	Channel A	Channel A
0	1	Channel B	Channel B
0	0	No Transmission	Ignore Frame

1) If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted resp. received frames are ignored (same function as CHA = CHB = 0)

FlexRay Protocol Controller (E-Ray)

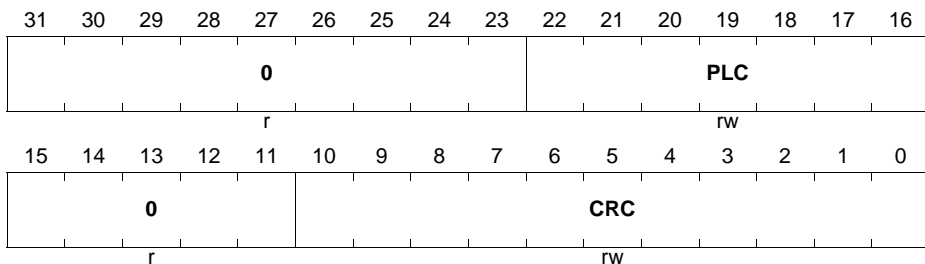
Write Header Section 2 (WRHS2)

WRHS2

Write Header Section 2

(0504_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
CRC	[10:0]	rw	Header CRC (vRF!Header!HeaderCRC) Receive Buffer: Configuration not required Transmit Buffer: Header CRC calculated and configured by the Host. For calculation of the header CRC the payload length of the frame send on the bus has to be considered. In static segment the payload length of all frames is configured by MHDC.SFDL .
PLC	[22:16]	rw	Payload Length Configured Length of data section (number of 2-byte words) as configured by the Host. During static segment the static frame payload length as configured by MHDC.SFDL in the MHD Configuration Register defines the payload length for all static frames. If the payload length configured by PLC is shorter than this value padding byte are inserted to ensure that frames have proper physical length. The padding pattern is logical zero.
0	[15:11], [31:23]	r	Reserved Returns 0 if read; should be written with 0.

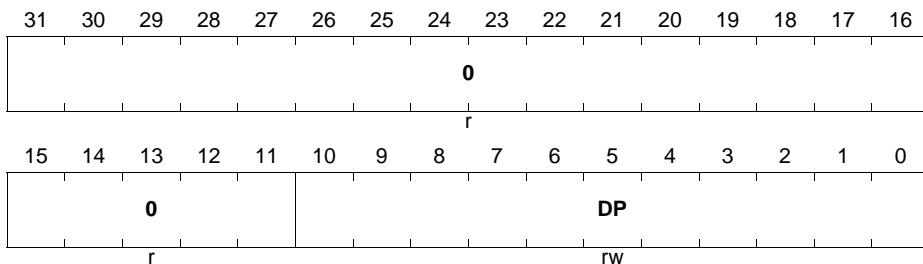
Write Header Section 3 (WRHS3)

WRHS3

Write Header Section 3

(0508_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
DP	[10:0]	rw	Data Pointer Pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.
0	[31:11]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

Input Buffer Command Mask (IBCM)

Configures how the message buffer in the Message RAM selected by the Input Buffer Command Request register **IBCR** is updated. When IBF Host and IBF Shadow are swapped, also masked bits **IBCM.LHSH**, **IBCM.LDSH**, and **IBCM.STXRH** are swapped with bits **IBCM.LHSS**, **IBCM.LDSS**, and **IBCM.STXRS** to keep them attached to the respective Input Buffer transfer.

IBCM

Input Buffer Command Mask (0510_H) **Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													STX RS	LD SS	LH SS
													rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													STX RH	LD SH	LH SH
													rw	rw	rw

Field	Bits	Type	Description
LHSH	0	rw	Load Header Section Host 0 _B Header section is not updated 1 _B Header section selected for transfer from Input Buffer to the Message RAM
LDSH	1	rw	Load Data Section Host 0 _B Data section is not updated 1 _B Data section selected for transfer from Input Buffer to the Message RAM
STXRH	2	rw	Set Transmission Request Host If this bit is set to 1, the Transmission Request flag TXRQ1.TXRn (n = 0-31) ... TXRQ4.TXRn (n = 0-31) for the selected message buffer is set in the Transmission Request Registers to release the message buffer for transmission. In single-shot mode the flag is cleared by the Communication Controller after transmission has completed. TXRQ1.TXRn (n = 0-31) ... TXRQ4.TXRn (n = 0-31) are evaluated for transmit buffer only. 0 _B Reset Transmission Request flag 1 _B Set Transmission Request flag, transmit buffer released for transmission

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
LHSS	16	rh	Load Header Section Shadow 0_B Header section is not updated 1_B Header section selected for transfer from Input Buffer to the Message RAM (transfer is ongoing of finalized)
LDSS	17	rh	Load Data Section Shadow 0_B Data section is not updated 1_B Data section selected for transfer from Input Buffer to the Message RAM (transfer is ongoing of finalized)
STXRS	18	rh	Transmission Request Shadow If this bit is set to 1, the Transmission Request flag TXRQ1.TXRn (n = 0-31) ... TXRQ4.TXRn (n = 0-31) for the selected message buffer is set in the Transmission Request Registers to release the message buffer for transmission. In single-shot mode the flag is cleared by the Communication Controller after transmission has completed. TXRQ1.TXRn (n = 0-31) ... TXRQ4.TXRn (n = 0-31) are evaluated for transmit buffer only. 0_B Reset Transmission Request flag 1_B Set Transmission Request flag, transmit buffer released for transmission (operation is ongoing of finalized)
0	[15:3], [31:19]	r	Reserved Returns 0 if read; should be written with 0.

Input Buffer Command Request (IBCR)

When the Host writes the number of the target message buffer in the Message RAM to IBRH in the Input Buffer Command Request register, IBF Host and IBF Shadow are swapped. In addition the message buffer numbers stored under IBRH and IBRS are also swapped (see also [“Data Transfer from Input Buffer to Message RAM” on Page 2-212](#)).

With this write operation the IBSYS bit in the Input Buffer Command Request register is set to 1. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by IBRS.

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message into the IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, the IBSYS bit is set back to 0 and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to IBRH.

If a write access to IBRH occurs while IBSYS is 1, IBSYH is set to 1. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, IBSYH is reset to 0. IBSYS remains set to 1, and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under IBRH and IBRS are also swapped. Any write access to an Input Buffer register while both IBSYS and IBSYH are set will cause the error flag [EIR.IBA](#) to be set. In this case the Input Buffer will not be changed.

IBCR

Input Buffer Command Request (0514_H) Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IB SYS		0								IBRS					
		r								rh					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IB SYH		0								IBRH					
		r								rw					

Field	Bits	Type	Description
IBRH	[6:0]	rw	Input Buffer Request Host Selects the target message buffer in the Message RAM for data transfer from Input Buffer. Valid values are 00 _H to 7F _H (0...127).

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
IBSYH	15	rh	Input Buffer Busy Host Set to 1 by writing IBRH while IBSYS is still 1. After the ongoing transfer between IBF Shadow and the Message RAM has completed, the IBSYH is set back to 0. 0 _B No request pending 1 _B Request while transfer between IBF Shadow and Message RAM in progress
IBRS	[22:16]	rh	Input Buffer Request Shadow Number of the target message buffer actually updated/lately updated. Valid values are 00 _H to 7F _H (0...127).
IBSYS	31	rh	Input Buffer Busy Shadow Set to 1 after writing IBRH. When the transfer between IBF Shadow and the Message RAM has completed, IBSYS is set back to 0. 0 _B Transfer between IBF Shadow and Message RAM completed 1 _B Transfer between IBF Shadow and Message RAM in progress
0	[14:7], [30:23]	r	Reserved Returns 0 if read; should be written with 0.

2.4.2.10 Output Buffer

Double buffer structure consisting of Output Buffer Host and Output Buffer Shadow. Used to read out message buffers from the Message RAM. While the Host can read from Output Buffer Host, the Message Handler transfers the selected message buffer from Message RAM to the respective Output Buffer Shadow. The data transfer between Message RAM and Output Buffer (OBF) is described in [“Data Transfer from Message RAM to Output Buffer” on Page 2-215](#).

Read Data Section [1...64] (RDDS_n)

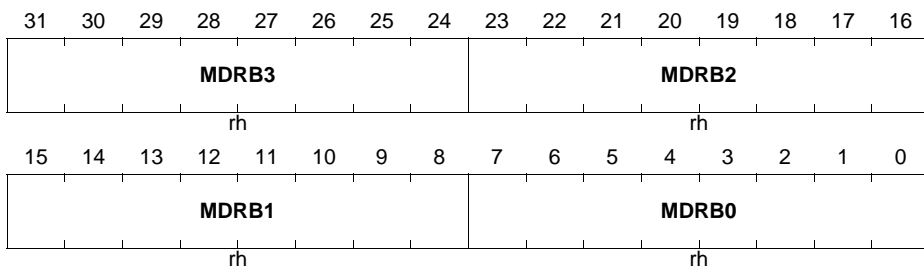
Holds the data words read from the data section of the addressed message buffer. The data words are read from the Message RAM in reception order from DW1 (byte0, byte1) to DW_{PL} (PL = number of data words as defined by the Payload Length).

RDDS_{nn} (nn = 01-64)

Read Data Section nn

$(05FC_H + nn * 4)$

Reset Value: 0000 0000_H



Field	Bits	Type	Description
MDRB0	[7:0]	rh	32-Bit Word nn, Byte 0
MDRB1	[15:8]	rh	32-Bit Word nn, Byte 1
MDRB2	[23:16]	rh	32-Bit Word nn, Byte 2
MDRB3	[31:24]	rh	32-Bit Word nn, Byte 3

Note: DW127 is located on RDDS64.MDW. In this case RDDS64.MDW is unused (no valid data). The Output Buffer RAMs are initialized to zero when leaving hard reset or by CHI command CLEAR_RAMs.

FlexRay Protocol Controller (E-Ray)

Read Header Section 1 (RDHS1)

Values as configured by the Host via **WRHS1** Register:

RDHS1

Read Header Section 1

(0700_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC							
r	rh	rh	rh	rh	rh	rh	r	rh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0					FID										
r					rh										

Field	Bits	Type	Description
FID	[10:0]	rh	Frame ID
CYC	[22:16]	rh	Cycle Code
CHA	24	rh	Channel Filter Control A
CHB	25	rh	Channel Filter Control B
CFG	26	rh	Message Buffer Direction Configuration Bit
PPIT	27	rh	Payload Preamble Indicator Transmit
TXM	28	rh	Transmission Mode
MBI	29	rh	Message Buffer Service Request
0	[15:11], 23, [31:30]	r	Reserved Returns 0 if read; should be written with 0.

Note: In case that the message buffer read from the Message RAM belongs to the receive FIFO, FID holds the received frame ID, while CYC, CHA, CHB, CFG, PPIT, TXM, and MBI are reset to zero.

Table 2-9 Channel Filter Control Bits

CHA	CHB	Transmit Buffer transmit frame on	Receive Buffer store frame received from
1 ¹⁾	1 ¹⁾	Both Channels (static segment only)	Channel A or B (store first semantically valid frame, static segment only)
1	0	Channel A	Channel A
0	1	Channel B	Channel B
0	0	No Transmission	Ignore Frame

1) If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted resp. received frames are ignored (same function as CHA = CHB = 0)

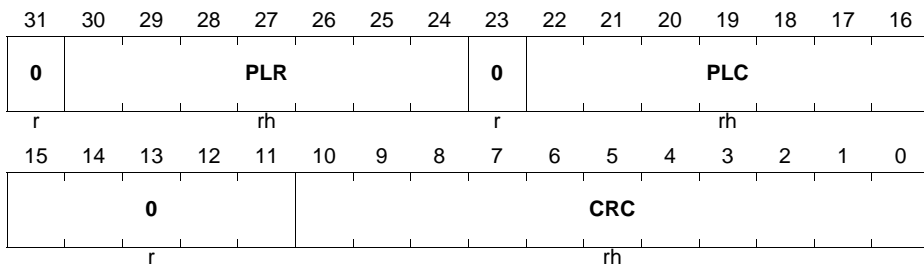
Read Header Section 2 (RDHS2)

RDHS2

Read Header Section 2

(0704_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
CRC	[10:0]	rh	Header CRC (vRF!Header!HeaderCRC) Receive Buffer: Configuration not required. Header CRC updated from receive data frames. Transmit Buffer: Header CRC calculated and configured by the Host
PLC	[22:16]	rh	Payload Length Configured Length of data section (number of 2-byte words) as configured by the Host.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
PLR	[30:24]	rh	Payload Length Received (vRF!Header!Length) Payload length value updated from received data frame (exception: if message buffer belongs to the receive FIFO PLR is also updated from received null frames). When a message is stored into a message buffer the following behavior with respect to payload length received and payload length configured is implemented: <ul style="list-style-type: none"> • PLR > PLC: The payload data stored in the message buffer is truncated to the payload length configured for even PLC or else truncated to PLC + 1. • PLR ≤ PLC: The received payload data is stored into the message buffers data section. The remaining data bytes of the data section as configured by PLC are filled with undefined data. • PLR = 0: The message buffer's data section is filled with undefined data. • PLC = 0: Message buffer has no data section configured. No data is stored into the message buffer's data section.
0	[15:11], 23, 31	r	Reserved Returns 0 if read; should be written with 0.

Note: The Message RAM is organized in 4-byte words. When received data is stored into a message buffer's data section, the number of 2-byte data words written into the message buffer is PLC rounded to the next even value. PLC should be configured identical for all message buffers belonging to the receive FIFO. Header 2 is updated from data frames only.

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
PPI	28	rh	Payload Preamble Indicator (vRF!Header!PPIndicator) The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame. 0 _B The payload segment of the received frame does not contain a network management vector nor a message ID 1 _B Static segment: Network management vector in the first part of the payload Dynamic segment: Message ID in the first part of the payload
RES	29	rh	Reserved Bit (vRF!Header!Reserved) Reflects the state of the received reserved bit. The reserved bit is transmitted as 0.
0	[15:11], [23:22], [31:30]	r	Reserved Returns 0 if read; should be written with 0.

Note: Header 3 is updated from data frames only.

Message Buffer Status (MBS)

The message buffer status is updated by the Communication Controller with respect to the assigned channel(s) latest at the end of the slot following the slot assigned to the message buffer. The flags are updated only when the Communication Controller is in "NORMAL_ACTIVE" or "NORMAL_PASSIVE" state. If only one channel (A or B) is assigned to a message buffer, the channel-specific status flags of the other channel are written to zero. If both channels are assigned to a message buffer, the channel-specific status flags of both channels are updated. The message buffer status is updated only when the slot counter reached the configured frame ID and when the cycle counter filter matched. When the Host updates a message buffer via Input Buffer, all MBS flags are reset to zero independent of which **IBCM** bits are set or not. For details about receive / transmit filtering see "**Filtering and Masking**" on Page 2-201, "**Transmit Process**" on Page 2-205, and "**Receive Process**" on Page 2-207.

Whenever the Message Handler changes one of the flags **VFRA**, **VFRB**, **SEOA**, **SEOB**, **CEOA**, **CEOB**, **SVOA**, **SVOB**, **TCIA**, **TCIB**, **ESA**, **ESB**, **MLST**, **FTA**, **FTB** the respective message buffer's MBC flag in registers **MBSC1** ... **MBSC4** is set

MBS

Message Buffer Status

(070C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	RES S	PPIS	NFIS	SYN S	SFIS	RCIS	0	CCS							
r	rh	rh	rh	rh	rh	rh	r	rh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTB	FTA	0	ML ST	ESB	ESA	TCIB	TCIA	SV OB	SV OA	CE OB	CE OA	SE OB	SE OA	VR FB	VR FA
rh	rh	r	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
VFRA	0	rh	Valid Frame Received on Channel A (vSS!ValidFrameA) A valid frame indication is set if a valid frame was received on channel A. 0 _B No valid frame received on channel A 1 _B Valid frame received on channel A
VFRB	1	rh	Valid Frame Received on Channel B (vSS!ValidFrameB) A valid frame indication is set if a valid frame was received on channel B. 0 _B No valid frame received on channel B 1 _B Valid frame received on channel B

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
SEOA	2	rh	Syntax Error Observed on Channel A (vSS!SyntaxErrorA) A syntax error was observed in the assigned slot on channel A. 0 _B No syntax error observed on channel A 1 _B Syntax error observed on channel A
SEOB	3	rh	Syntax Error Observed on Channel B (vSS!SyntaxErrorB) A syntax error was observed in the assigned slot on channel B. 0 _B No syntax error observed on channel B 1 _B Syntax error observed on channel B
CEOA	4	rh	Content Error Observed on Channel A (vSS!ContentErrorA) A content error was observed in the assigned slot on channel A. 0 _B No content error observed on channel A 1 _B Content error observed on channel A
CEOB	5	rh	Content Error Observed on Channel B (vSS!ContentErrorB) A content error was observed in the assigned slot on channel B. 0 _B No content error observed on channel B 1 _B Content error observed on channel B
SVOA	6	rh	Slot Boundary Violation Observed on Channel A (vSS!BViolationA) A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel A. 0 _B No slot boundary violation observed on channel A 1 _B Slot boundary violation observed on channel A
SVOB	7	rh	Slot Boundary Violation Observed on Channel B (vSS!BViolationB) A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel B. 0 _B No slot boundary violation observed on channel B 1 _B Slot boundary violation observed on channel B
TCIA	8	rh	Transmission Conflict Indication Channel A (vSS!TxConflictA) A transmission conflict indication is set if a transmission conflict has occurred on channel A. 0 _B No transmission conflict occurred on channel A 1 _B Transmission conflict occurred on channel A

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
TCIB	9	rh	Transmission Conflict Indication Channel B (vSS!TxConflictB) A transmission conflict indication is set if a transmission conflict has occurred on channel B. 0 _B No transmission conflict occurred on channel B 1 _B Transmission conflict occurred on channel B
ESA	10	rh	Empty Slot Channel A In an empty slot there is no activity detected on the bus. The condition is checked in static and dynamic slots. 0 _B Bus activity detected in the assigned slot on channel A 1 _B No bus activity detected in the assigned slot on channel A
ESB	11	rh	Empty Slot Channel B In an empty slot there is no activity detected on the bus. The condition is checked in static and dynamic slots. 0 _B Bus activity detected in the assigned slot on channel B 1 _B No bus activity detected in the assigned slot on channel B
MLST	12	rh	Message Lost The flag is set in case the Host did not read the message before the message buffer was updated from a received data frame. Not affected by reception of null frames except for message buffers belonging to the receive FIFO. The flag is reset by a Host write to the message buffer via IBF or when a new message is stored into the message buffer after the message buffers ND flag was reset by reading out the message buffer via OBF. 0 _B No message lost 1 _B Unprocessed message was overwritten
FTA	14	rh	Frame Transmitted on Channel A Indicates that this node has transmitted a data frame in the assigned slot on channel A. 0 _B No transmission transmitted on channel A 1 _B Data frame transmitted on channel A in cycle defined by CCS bitfield <i>Note: The FlexRay protocol specification requires that FTA can only be reset by the Host. Therefore the Cycle Count Status CCS for these bits is only valid for the cycle where the bits are set to 1</i>

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
FTB	15	rh	<p>Frame Transmitted on Channel B</p> <p>Indicates that this node has transmitted a data frame in the assigned slot on channel B.</p> <p>0_B No transmission transmitted on channel B</p> <p>1_B Data frame transmitted on channel B in cycle defined by CCS bitfield</p> <p><i>Note: The FlexRay protocol specification requires that FTB can only be reset by the Host. Therefore the Cycle Count Status CCS for these bits is only valid for the cycle where the bits are set to 1</i></p>
CCS	[21:16]	rh	<p>Cycle Count Status</p> <p>Cycle Count when status (MBS register) has been updated.</p>
RCIS	24	rh	<p>Received on Channel Indicator Status (vSS!Channel)</p> <p>Indicates the channel on which the frame was received.</p> <p>0_B Frame received on channel B</p> <p>1_B Frame received on channel A</p> <p><i>Note: For receive buffers (CFG = 0) the RCIS is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flags have no meaning and should be ignored.</i></p>
SFIS	25	rh	<p>Startup Frame Indicator Status (vRF!Header!SuFIndicator)</p> <p>A startup frame is marked by the startup frame indicator.</p> <p>0_B No startup frame received</p> <p>1_B The received frame is a startup frame</p> <p><i>Note: For receive buffers (CFG = 0) the SFIS is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flags have no meaning and should be ignored.</i></p>
SYNS	26	rh	<p>Sync Frame Indicator Status (vRF!Header!SyFIndicator)</p> <p>A startup frame is marked by the startup frame indicator.</p> <p>0_B No sync frame received</p> <p>1_B The received frame is a sync frame</p> <p><i>Note: For receive buffers (CFG = 0) the SYNS is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flags have no meaning and should be ignored.</i></p>

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
NFIS	27	rh	<p>Null Frame Indicator Status (vRF!Header!NFIndicator)</p> <p>If reset to 0 the payload segment of the received frame contains no usable data.</p> <p>0_B Received frame is a null frame</p> <p>1_B Received frame is not a null frame</p> <p><i>Note: For receive buffers (CFG = 0) the NFIS is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flags have no meaning and should be ignored.</i></p>
PPIS	28	rh	<p>Payload Preamble Indicator Status (vRF!Header!PPIndicator)</p> <p>The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.</p> <p>Static Segment:</p> <p>0_B The payload segment of the received frame does not contain a network management vector or a message ID</p> <p>1_B Network management vector at the beginning of the payload</p> <p>Dynamic Segment:</p> <p>0_B The payload segment of the received frame does not contain a network management vector or a message ID</p> <p>1_B Message ID at the beginning of the payload</p> <p><i>Note: For receive buffers (CFG = 0) the PPIS is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flags have no meaning and should be ignored.</i></p>
RESS	29	rh	<p>Reserved Bit Status (vRF!Header!Reserved)</p> <p>Reflects the state of the received reserved bit. The reserved bit is transmitted as 0.</p> <p><i>Note: For receive buffers (CFG = 0) the RESS is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flags have no meaning and should be ignored.</i></p>
0	13, [23:22], [31:30]	r	<p>Reserved</p> <p>Returns 0 if read; should be written with 0.</p>

FlexRay Protocol Controller (E-Ray)

Output Buffer Command Mask (OBCM)

Configures how the Output Buffer is updated from the message buffer in the Message RAM selected by the Output Buffer Command Request register. When OBF Host and OBF Shadow are swapped, also mask bits **OBCM.RDSH** and **OBCM.RHSH** are swapped with bits **OBCM.RDSS** and **OBCM.RHSS** to keep them attached to the respective Output Buffer transfer.

OBCM

Output Buffer Command Mask (0710_H) **Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0														RD SH	RH SH
														rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														RD SS	RH SS
														rw	rw

Field	Bits	Type	Description
RHSS	0	rw	Read Header Section Shadow 0 _B Header section is not read 1 _B Header section selected for transfer from Message RAM to Output Buffer
RDSS	1	rw	Read Data Section Shadow 0 _B Data section is not read 1 _B Data section selected for transfer from Message RAM to Output Buffer
RHSH	16	rh	Read Header Section Host 0 _B Header section is not read 1 _B Header section selected for transfer from Message RAM to Output Buffer
RDSH	17	rh	Read Data Section Host 0 _B Data section is not read 1 _B Data section selected for transfer from Message RAM to Output Buffer
0	[15:2], [31:18]	r	Reserved Returns 0 if read; should be written with 0.

FlexRay Protocol Controller (E-Ray)

*Note: After the transfer of the header section from the Message RAM to OBF Shadow has completed, the message buffer status changed flag **MBCn** ($n = 0-31$) ... **MBCn** ($n = 96-127$) of the selected message buffer in the Message Buffer Changed **MBSC1** ... **MBSC4** registers is cleared. After the transfer of the data section from the Message RAM to OBF Shadow has completed, the New Data flag **NDn** ($n = 0-31$) ... **NDn** ($n = 96-127$) of the selected message buffer in the New Data **NDAT1** ... **NDAT4** registers is cleared.*

Output Buffer Command Request (OBCR)

The message buffer selected by **OBCR.OBRS** is transferred from the Message RAM to the Output Buffer as soon as the Host has set **OBCR.REQ** to 1. Bit **OBCR.REQ** can only be set to 1 while **OBCR.OBSYS** is 0 (see also “Data Transfer from Message RAM to Output Buffer” on Page 2-215).

After setting **OBCR.REQ** to 1, **OBCR.OBSYS** is automatically set to 1, and the transfer of the message buffer selected by **OBCR.OBRS** from the Message RAM to OBF Shadow is started. When the transfer between the Message RAM and OBF Shadow has completed, this is signalled by setting **OBCR.OBSYS** back to 0. By setting the **OBCR.VIEW** bit to 1 while **OBCR.OBSYS** is 0, OBF Host and OBF Shadow are swapped. Now the Host can read the transferred message buffer from OBF Host. In parallel the Message Handler may transfer the next message from the Message RAM to OBF Shadow if **OBCR.VIEW** and **OBCR.REQ** are set at the same time.

Any write access to an Output Buffer register while **OBCR.OBSYS** is set will cause the error flag **EIR.IOBA** to be set. In this case the Output Buffer will not be changed.

OBCR

Output Buffer Command Request (0714_H) Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0									OBRH						
r									rh						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OBSYS	0					REQ	VIEW	0	OBRS						
rh	r					rw	rw	r	rw						

FlexRay Protocol Controller (E-Ray)

Field	Bits	Type	Description
OBRS	[6:0]	rw	Output Buffer Request Shadow Number of source message buffer to be transferred from the Message RAM to OBF Shadow. Valid values are 00 _H to 7F _H (0...127). If the number of the first message buffer of the receive FIFO is written to this register the Message Handler transfers the message buffer addressed by the GET Index Register (GIDX, "FIFO Function" on Page 2-209) to OBF Shadow.
VIEW	8	rw	View Shadow Buffer Toggles between OBF Shadow and OBF Host. Only writeable while OBCR.OBSYS = 0. 0 _B No action 1 _B Swap OBF Shadow and OBF Host
REQ	9	rw	Request Message RAM Transfer Requests transfer of message buffer addressed by OBCR.OBRS from Message RAM to OBF Shadow. Only writeable while OBCR.OBSYS = 0. 0 _B No request 1 _B Transfer to OBF Shadow requested
OBSYS	15	rh	Output Buffer Busy Shadow Set to 1 after setting bit OBCR.REQ . When the transfer between the Message RAM and OBF Shadow has completed, OBCR.OBSYS is set back to 0. 0 _B No transfer in progress 1 _B Transfer between Message RAM and OBF Shadow in progress
OBRRH	[22:16]	rh	Output Buffer Request Host Number of message buffer currently accessible by the Host via RDHS1...RDHS3 , MBS , and RDDSnn (nn = 01-64) . By writing OBCR.VIEW to 1 OBF Shadow and OBF Host are swapped and the transferred message buffer is accessible by the Host. Valid values are 00 _H to 7F _H (0...127).
0	7, [14:10], [31:23]	r	Reserved Returns 0 if read; should be written with 0.

2.5 Functional Description

This chapter describes the E-Ray implementation together with the related FlexRay protocol features. More information about the FlexRay protocol itself can be found in the FlexRay protocol specification v2.1.

Communication on FlexRay networks is based on frames and symbols. The wakeup symbol (WUS) and the collision avoidance symbol (CAS) are transmitted outside the communication cycle to setup the time schedule. Frames and media access test symbols (MTS) are transmitted inside the communication cycle.

2.5.1 Communication Cycle

A communication cycle in FlexRay consists of the following elements:

- Static Segment
- Dynamic Segment
- Symbol Window
- Network Idle Time (NIT)

Static segment, dynamic segment, and symbol window form the Network Communication Time (NCT). For each communication channel the slot counter starts at 1 and counts up until the end of the dynamic segment is reached. Both channels share the same arbitration grid which means that they use the same synchronized macrotick.

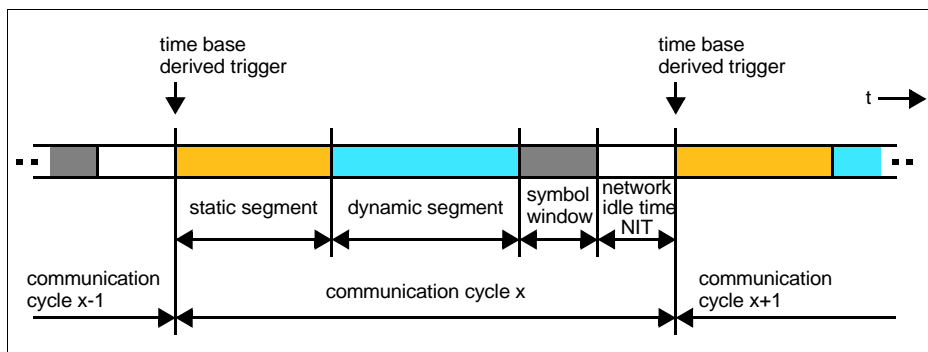


Figure 2-3 Structure of Communication Cycle

2.5.1.1 Static Segment

The Static Segment is characterized by the following features:

- Time slots of fixed length (optionally protected by bus guardian)
- Start of frame transmission at action point of the respective static slot
- Payload length same for all frames on both channel

FlexRay Protocol Controller (E-Ray)

Parameters: Number of Static Slots **GTUC07.NSS**, Static Slot Length **GTUC07.SSL**, Payload Length Static **MHDC.SFDL**, Action Point Offset **GTUC09.APO**.

2.5.1.2 Dynamic Segment

The Dynamic Segment is characterized by the following features:

- All controllers have bus access (no bus guardian protection possible)
- Variable payload length and duration of slots, different for both channels
- Start of transmission at minislot action point

Parameters: Number of Minislots **GTUC08.NMS**, Minislot Length **GTUC08.MSL**, Minislot Action Point Offset **GTUC09.MAPO**, Start of Latest Transmit (last minislot) **MHDC.SLT**.

2.5.1.3 Symbol Window

During the symbol window only one media access test symbol (MTS) may be transmitted per channel. MTS symbols are sent in "NORMAL_ACTIVE" state to test the bus guardian.

The symbol window is characterized by the following features:

- Send single symbol
- Transmission of the MTS symbol starts at the symbol windows action point

Parameters: Symbol Window Action Point Offset **GTUC09.APO** (same as for static slots), Network Idle Time Start **GTUC04.NIT**.

2.5.1.4 Network Idle Time (NIT)

During network idle time the Communication Controller has to perform the following tasks:

- Calculate clock correction terms (offset and rate)
- Distribute offset correction over multiple macroticks
- Perform cluster cycle related tasks

Parameters: Network Idle Time Start **GTUC04.NIT**, Offset Correction Start **GTUC04.OCS**.

2.5.1.5 Configuration of Network Idle Time (NIT) Start and Offset Correction Start

The number of macroticks per cycle (gMacroPerCycle) is assumed to be m. It is configured by programming **GTUC02.MPC** = m.

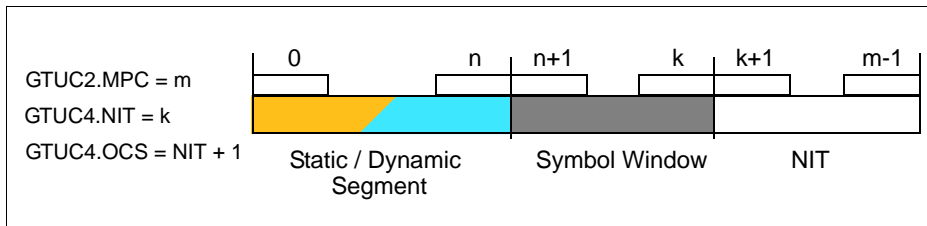


Figure 2-4 Configuration of network idle time (NIT) start and offset correction start

The static / dynamic segment starts with macrotick 0 and ends with macrotick n:

$n = \text{static segment length} + \text{dynamic segment offset} + \text{dynamic segment length} - 1 \text{ macrotick}$

$n = \text{gNumberOfStaticSlots} \cdot \text{gdStaticSlot} + \text{dynamic segment offset} + \text{gNumberOfMinislots} \cdot \text{gdMinislot} - 1 \text{ macroticks}$

The static segment length is configured by **GTUC07.SSL** and **GTUC07.NSS**.

The dynamic segment length is configured by **GTUC08.MSL** and **GTUC08.NMS**.

The dynamic segment offset is:

If $\text{gdActionPointOffset} \leq \text{gdMinislotActionPointOffset}$:

dynamic segment offset = 0 MT

Else if $\text{gdActionPointOffset} > \text{gdMinislotActionPointOffset}$:

dynamic segment offset = $\text{gdActionPointOffset} - \text{gdMinislotActionPointOffset}$

The network idle time (NIT) starts with macrotick k+1 and ends with the last macrotick of cycle m-1. It has to be configured by setting **GTUC04.NIT** = k.

For the E-Ray the offset correction start is required to be

GTUC04.OCS \geq **GTUC04.NIT** + 1 = k+1.

The length of symbol window results from the number of macroticks between the end of the static / dynamic segment and the beginning of the NIT. It can be calculated by $k - n$.

2.5.2 Communication Modes

The FlexRay Protocol Specification v2.1 defines the Time-Triggered Distributed (TT-D) mode.

Time-triggered Distributed (TT-D)

In TT-D mode the following configurations are possible:

- **Pure static:** minimum 2 static slots + symbol window (optional)
- **Mixed static/dynamic:** minimum 2 static slots + dynamic segment + symbol window (optional)

A minimum of two coldstart nodes need to be configured for distributed time-triggered operation. Two fault-free coldstart nodes are necessary for the cluster startup. Each startup frame must be a sync frame, therefore all coldstart nodes are sync nodes.

2.5.3 Clock Synchronization

In TT-D mode a distributed clock synchronization is used. Each node individually synchronizes itself to the cluster by observing the timing of received sync frames from other nodes.

2.5.3.1 Global Time

Activities in a FlexRay node, including communication, are based on the concept of a global time, even though each individual node maintains its own view of it. It is the clock synchronization mechanism that differentiates the FlexRay cluster from other node collections with independent clock mechanisms. The global time is a vector of two values; the cycle (cycle counter) and the cycle time (macrotick counter).

Cluster specific:

- Macrotick = basic unit of time measurement in a FlexRay network, a macrotick consists of an integer number of microticks
- Cycle length = duration of a communication cycle in units of macroticks

2.5.3.2 Local Time

Internally, nodes time their behavior with microtick resolution. Microticks are time units derived from the oscillator clock tick of the specific node. Therefore microticks are controller-specific units. They may have different duration in different controllers. The precision of a node's local time difference measurements is a microtick.

Node specific:

- Oscillator clock → prescaler → microtick
- microtick = basic unit of time measurement in a Communication Controller, clock correction is done in units of microticks
- Cycle counter + macrotick counter = nodes local view of the global time

2.5.3.3 Synchronization Process

Clock synchronization is performed by means of sync frames. Only preconfigured nodes (sync nodes) are allowed to send sync frames. In a two-channel cluster a sync node has to send its sync frame on both channels.

For synchronization in FlexRay the following constraints have to be considered:

- Max. one sync frame per node in one communication cycle
- Max. 15 sync frames per cluster in one communication cycle
- Every node has to use all available sync frames for clock synchronization
- Minimum of two sync nodes required for clock synchronization and startup

For clock synchronization the time difference between expected and observed arrival time of sync frames received during the static segment, valid on both channels (two-channel cluster), is measured. The calculation of correction terms is done during network idle time (NIT) (offset: every cycle, rate: odd cycle) by using a FTA / FTM algorithm. For details see FlexRay protocol specification v2.1, chapter 8.

Offset (phase) Correction

- Only deviation values measured and stored in the current cycle used
- For a two channel node the smaller value will be taken
- Calculation during network idle time (NIT) of **every** communication cycle, value may be negative
- Offset correction value calculated in even cycles used for error checking only
- Checked against limit values (violation: "NORMAL_ACTIVE" → "NORMAL_PASSIVE" → "HALT")
- Correction value is an integer number of microticks
- Correction done in **odd** numbered cycles, distributed over the macroticks beginning at offset correction start up to cycle end (end of network idle time (NIT)) to shift nodes next start of cycle (macroticks lengthened / shortened)

Synchronization Process

Clock synchronization is performed by means of sync frames. Only preconfigured nodes (sync nodes) are allowed to send sync frames. In a two-channel cluster a sync node has to send its sync frame on both channels.

FlexRay Protocol Controller (E-Ray)

For synchronization in FlexRay the following constraints have to be considered:

- Max. one sync frame per node in one communication cycle
- Max. 15 sync frames per cluster in one communication cycle
- Every node has to use all available sync frames for clock synchronization
- Minimum of two sync nodes required for clock synchronization and startup

For clock synchronization the time difference between expected and observed arrival time of sync frames received during the static segment, valid on both channels (two-channel cluster), is measured. The calculation of correction terms is done during network idle time (NIT) (offset: every cycle, rate: odd cycle) by using a FTA / FTM algorithm. For details see FlexRay protocol specification v2.1, chapter 8.

Offset (phase) Correction

- Only deviation values measured and stored in the current cycle used
- For a two channel node the smaller value will be taken
- Calculation during network idle time (NIT) of **every** communication cycle, value may be negative
- Offset correction value calculated in even cycles used for error checking only
- Checked against limit values (violation: "NORMAL_ACTIVE" → "NORMAL_PASSIVE" → "HALT")
- Correction value is an integer number of microticks
- Correction done in **odd** numbered cycles, distributed over the macroticks beginning at offset correction start up to cycle end (end of network idle time (NIT)) to shift nodes next start of cycle (macroticks lengthened / shortened)

Rate (frequency) Correction

- Pairs of deviation values measured and stored in even / odd cycle pair used
- For a two channel node the average of the differences from the two channels is used
- Calculated during network idle time (NIT) of **odd** numbered cycles, value may be negative
- Cluster drift damping is performed using global damping value
- Checked against limit values
- Correction value is a signed integer number of microticks
- Distributed over macroticks comprising the next **even / odd** cycle pair (macroticks lengthened / shortened)

Sync Frame Transmission

Sync frame transmission is only possible from buffer 0 and 1. Message buffer 1 may be used for sync frame transmission in case that sync frames should have different payloads on the two channels. In this case bit **MRC.SPLM** has to be programmed to 1.

FlexRay Protocol Controller (E-Ray)

Message buffers used for sync frame transmission have to be configured with the key slot ID and can be (re)configured in “DEFAULT_CONFIG” or “CONFIG” state only. For nodes transmitting sync frames **SUCC1.TXSY** must be set to 1.

2.5.3.4 External Clock Synchronization

During normal operation, independent clusters can drift significantly. If synchronous operation across independent clusters is desired, external synchronization is necessary; even though the nodes within each cluster are synchronized. This can be accomplished with synchronous application of host-deduced rate and offset correction terms to the clusters.

- External offset / rate correction value is a signed integer
- External offset / rate correction value is added to calculated offset / rate correction value
- Aggregated offset / rate correction term (external + internal) is not checked against configured limits

2.5.4 Error Handling

The implemented error handling concept is intended to ensure that in case of a lower layer protocol error in a single node communication between non-affected nodes can be maintained. In some cases, higher layer program command activity is required for the Communication Controller to resume normal operation. A change of the error handling state will set bit **EIR.PEMC** in the Error Service Request Register and may trigger an service request to the Host if enabled. The actual error mode is signalled by **CCEV.ERRM** in the Communication Controller Error Vector register.

Table 2-10 Error Modes of the POC (Degradation Model)

Error Mode	Activity
ACTIVE (green)	Full operation , State: "NORMAL_ACTIVE" The Communication Controller is fully synchronized and supports the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers EIR and SIR .
PASSIVE (yellow)	Reduced operation , State: "NORMAL_PASSIVE", Communication Controller self rescue allowed The Communication Controller stops transmitting frames and symbols, but received frames are still processed. Clock synchronization mechanisms are continued based on received frames. No active contribution to the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers EIR and SIR .
COMM_HALT (red)	Operation halted , State: "HALT", Communication Controller self rescue not allowed The Communication Controller stops frame and symbol processing, clock synchronization processing, and the macrotick generation. The host has still access to error and status information by reading the error and status interrupt flags from registers EIR and SIR . The bus drivers are disabled.

2.5.4.1 Clock Correction Failed Counter

When the Clock Correction Failed Counter reaches the maximum "without clock correction passive" limit defined by **SUCC3.WCP**, the POC transits from "NORMAL_ACTIVE" to "NORMAL_PASSIVE" state. When it reaches the "maximum without clock correction fatal" limit defined by **SUCC3.WCF**, it transits "NORMAL_ACTIVE" or "NORMAL_PASSIVE" to the "HALT" state. Both limits are defined in the SUC Configuration Register 3.

The Clock Correction Failed Counter **CCEV.CCFC** allows the Host to monitor the duration of the inability of a node to compute clock correction terms after the Communication Controller passed protocol startup phase. It will be incremented by one at the end of any **odd** numbered communication cycle where either the Missing Offset Correction signal **SFS.MOCS** nor the Missing Rate Correction signal **SFS.MRCS** flag is set. The two flags are located in the Sync Frame Status register, while the Clock Correction Failed Counter is located in the Communication Controller Error Vector register.

FlexRay Protocol Controller (E-Ray)

The Clock Correction Failed Counter is reset to zero at the end of an **odd** communication cycle if neither the Missing Offset Correction signal **SFS.MOCS** nor the Missing Rate Correction signal **SFS.MRCS** flag is set.

The Clock Correction Failed Counter stops incrementing when the “maximum without clock correction fatal” value **SUCC3.WCF** as defined in the SUC Configuration Register 3 is reached (i.e. incrementing the counter at its maximum value will not cause it to “wraparound” back to zero). The Clock Correction Failed Counter is initialized to zero when the Communication Controller enters “READY” state or when “NORMAL_ACTIVE” state is entered.

2.5.4.2 Passive to Active Counter

The passive to active counter controls the transition of the POC from “NORMAL_PASSIVE” to “NORMAL_ACTIVE” state. **SUCC1.PTA** in the SUC Configuration Register 1 defines the number of consecutive even / odd cycle pairs that must have valid clock correction terms before the Communication Controller is allowed to transit from “NORMAL_PASSIVE” to “NORMAL_ACTIVE” state. If **SUCC1.PTA** is reset to zero the Communication Controller is not allowed to transit from “NORMAL_PASSIVE” to “NORMAL_ACTIVE” state.

2.5.4.3 HALT Command

In case the Host wants to stop FlexRay communication of the local node it can bring the Communication Controller into “HALT” state by asserting the HALT command. This can be done by writing **SUCC1.CMD** = 0110_B in the SUC Configuration Register 1. When called in “NORMAL_ACTIVE” or “NORMAL_PASSIVE” state the POC transits to “HALT” state at the end of the current cycle. When called in any other state **SUCC1.CMD** will be reset to 0000_B = “COMMAND_NOT_ACCEPTED” and bit **EIR.CNA** in the Error Service Request Register is set to 1. If enabled an service request to the Host is generated.

2.5.4.4 FREEZE Command

In case the Host detects a severe error condition it can bring the Communication Controller into “HALT” state by asserting the FREEZE command. This can be done by writing **SUCC1.CMD** = 0111_B in the SUC Configuration Register 1. The FREEZE command triggers the entry of the “HALT” state immediately regardless of the actual POC state.

The POC state from which the transition to HALT state took place can be read from **CCSV.PSL**.

2.5.5 Communication Controller States

This chapter introduces the states of the Communication Controller.

2.5.5.1 Communication Controller State Diagram

State transitions are controlled by external pins `eray_reset` and `eray_rxd1,2`, by the POC state machine, and by the CHI Command Vector **SUCC1.CMD** located in the SUC Configuration Register 1.

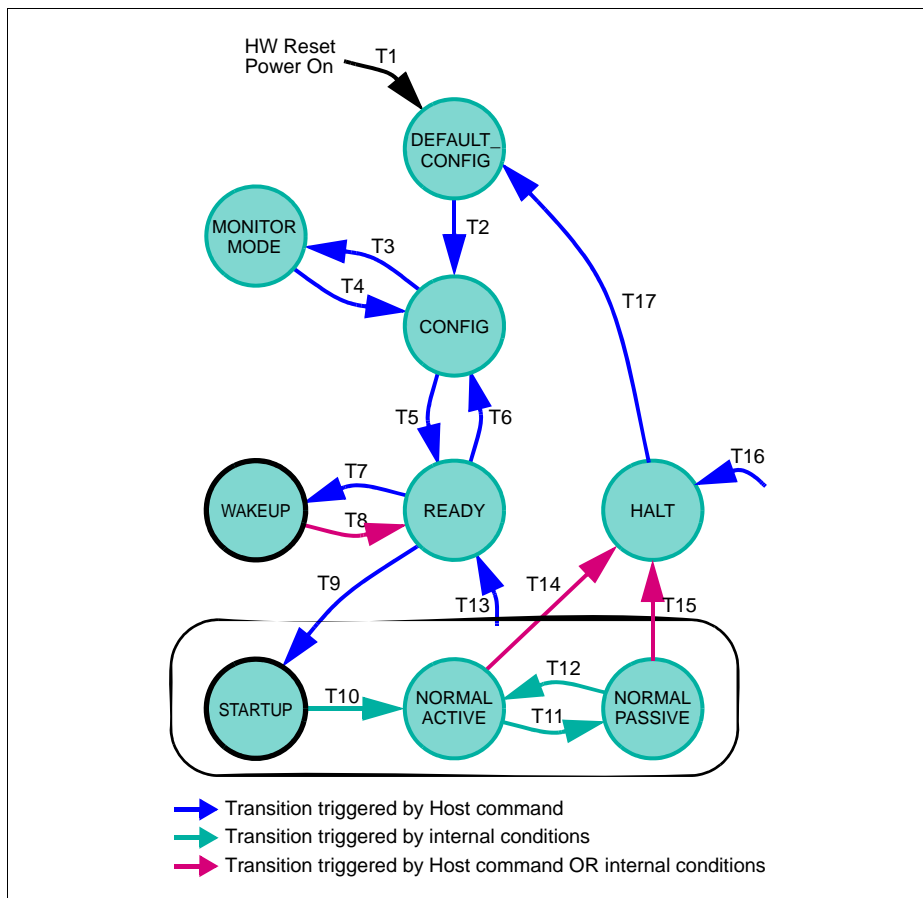


Figure 2-5 Overall State Diagram of E-Ray Communication Controller

The Communication Controller exits from all states to “HALT” state after application of the FREEZE command (**SUCC1.CMD**= 0111_B).

Table 2-11 State Transitions of E-Ray Overall State Machine

T#	Condition	From	To
1	Hardware reset	HW Reset	DEFAULT_CONFIG
2	Command CONFIG, SUCC1.CMD = 0001 _B	DEFAULT_CONFIG	CONFIG
3	Unlock sequence followed by command MONITOR_MODE, SUCC1.CMD = 1011 _B	CONFIG	MONITOR_MODE
4	Command CONFIG, SUCC1.CMD = 0001 _B	MONITOR_MODE	CONFIG
5	Unlock sequence followed by command READY, SUCC1.CMD = 0010 _B	CONFIG	READY
6	Command CONFIG, SUCC1.CMD = 0001 _B	READY	CONFIG
7	Command WAKEUP, SUCC1.CMD = 0011 _B	READY	WAKEUP
8	Complete, non-aborted transmission of wakeup pattern OR received WUP OR received frame header OR command READY, SUCC1.CMD = 0010 _B	WAKEUP	READY
9	Command RUN, SUCC1.CMD = 0100 _B	READY	STARTUP
10	Successful startup	STARTUP	NORMAL_ACTIVE
11	Clock Correction Failed counter reached Maximum Without Clock Correction Passive limit configured by WCP in SUC Configuration Register 3	NORMAL_ACTIVE	NORMAL_PASSIVE
12	Number of valid correction terms reached the Passive to Active limit configured by PTA in SUC Configuration Register 1	NORMAL_PASSIVE	NORMAL_ACTIVE
13	Command READY, SUCC1.CMD = 0010 _B	STARTUP, NORMAL_ACTIVE, NORMAL_PASSIVE	READY

FlexRay Protocol Controller (E-Ray)

Table 2-11 State Transitions of E-Ray Overall State Machine (cont'd)

T#	Condition	From	To
14	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by WCF in SUC Configuration Register 3 AND bit HCSE in the SUC Configuration Register 1 set to 1 OR command HALT, SUCC1.CMD = 0110 _B	NORMAL_ACTIVE	HALT
15	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by WCF in SUC Configuration Register 3 AND bit HCSE in the SUC Configuration Register 1 set to 1 OR command HALT, SUCC1.CMD = 0110 _B	NORMAL_PASSIVE	HALT
16	Command FREEZE, SUCC1.CMD = 0111 _B	All States	HALT
16	Command CONFIG, SUCC1.CMD = 0001 _B	HALT	DEFAULT_CONFIG

2.5.5.2 DEFAULT_CONFIG State

In “DEFAULT_CONFIG” state, the Communication Controller is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state.

The Communication Controller enters this state

- When leaving hardware reset (external reset signal eray_reset is disasserted)
- When exiting from “HALT” state

To leave “DEFAULT_CONFIG” state the Host has to write **SUCC1.CMD** = 0001_B in the SUC Configuration Register 1. The Communication Controller transits to “CONFIG” state.

CONFIG State

In “CONFIG” state, the Communication Controller is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state. This state is used to initialize the Communication Controller configuration.

FlexRay Protocol Controller (E-Ray)

The Communication Controller enters this state

- When exiting from “DEFAULT_CONFIG” state
- When exiting from “MONITOR_MODE” or “READY” state

When the state has been entered via “HALT” and “DEFAULT_CONFIG” state, the Host can analyze status information and configuration. Before leaving “CONFIG” state the Host has to assure that the configuration is fault-free.

To leave “CONFIG” state, the Host has to perform the unlock sequence as described on **“LCK” on Page 2-26**. Directly after unlocking the “CONFIG” state the Host has to write **SUCC1.CMD** in the SUC Configuration Register 1 to enter the next state.

Internal counters and the Communication Controller status flags are reset when the Communication Controller leaves “CONFIG”.

*Note: The Message Buffer Status Registers (**MHDS**, **TXRQ1** ... **TXRQ4**, **NDAT1** ... **NDAT4**, **MBSC1** ... **MBSC4**) and status data stored in the Message RAM and are not affected by the transition of the POC from “CONFIG” to “READY” state.*

When the Communication Controller is in “CONFIG” state it is also possible to bring the Communication Controller into a power saving mode by halting the module clocks (f_{SCLK} , $f_{\text{CLC_ERAY}}$). To do this the Host has to assure that all Message RAM transfers have finished before turning off the clocks.

2.5.5.3 MONITOR_MODE

After unlocking “CONFIG” state and writing **SUCC1.CMD** = 0011_B the Communication Controller enters “MONITOR_MODE”. In this mode the Communication Controller is able to receive FlexRay frames and to detect wakeup pattern. The temporal integrity of received frames is not checked, and therefore cycle counter filtering is not supported. This mode can be used for debugging purposes in case e.g. that startup of a FlexRay network fails. After writing **SUCC1.CMD** = 0001_B the Communication Controller transits back to “CONFIG” state.

In MONITOR_MODE the pick first valid mechanism is disabled. This means that a receive message buffer may only be configured to receive on one channel. Received frames are stored into message buffers according to frame ID and receive channel. Null frames are handled like data frames. After frame reception only status bits **MBS.VFRA**, **MBS**, **MBS.MLST**, **MBS.RCIS**, **MBS.SFIS**, **MBS.SYNS**, **MBS.NFIS**, **MBS.PPIS**, **MBS.RESS** have valid value. In MONITOR_MODE the receive FIFO is not available.

2.5.5.4 READY State

After unlocking “CONFIG” state and writing **SUCC1.CMD** = 0010_B the Communication Controller enters “READY” state. From this state the Communication Controller can transit to WAKEUP state and perform a cluster wakeup or to “STARTUP” state to perform a coldstart or to integrate into a running communication.

FlexRay Protocol Controller (E-Ray)

The Communication Controller enters this state

- When exiting from “CONFIG”, “WAKEUP”, “STARTUP”, “NORMAL_ACTIVE”, or “NORMAL_PASSIVE” state by writing **SUCC1.CMD** = 0010_B (READY command).

The Communication Controller exits from this state

- To “CONFIG” state by writing **SUCC1.CMD** = 0001_B (CONFIG command)
- To “WAKEUP” state by writing **SUCC1.CMD** = 0011_B (WAKEUP command)
- To “STARTUP” state by writing **SUCC1.CMD** = 0100_B (RUN command)

Internal counters and the Communication Controller status flags are reset when the Communication Controller enters “STARTUP” state.

*Note: Status bits **MHDS**, registers **TXRQ1** ... **TXRQ4**, and status data stored in the Message RAM are not affected by the transition of the POC from “READY” to “STARTUP” state.*

2.5.5.5 WAKEUP State

The description below is intended to help configuring wakeup for the E-Ray IP-module. A detailed description of the wakeup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.1.

The Communication Controller enters this state

- When exiting from “READY” state by writing **SUCC1.CMD** = 0011_B (WAKEUP command).

The Communication Controller exits from this state to “READY” state

- After complete non-aborted transmission of wakeup pattern
- After WUP reception
- After detecting a WUP collision
- After reception of a frame header
- By writing **SUCC1.CMD** = 0010_B (READY command)

The cluster wakeup must precede the communication startup in order to ensure that all mechanisms defined for the startup work properly. The minimum requirement for a cluster wakeup is that all bus drivers are supplied with power. A bus driver has the ability to wake up the other components of its node when it receives a wakeup pattern on its channel. At least one node in the cluster needs an **external** wakeup source.

The Host completely controls the wakeup procedure. It is informed about the state of the cluster by the bus driver and the Communication Controller and configures bus guardian (if available) and Communication Controller to perform the cluster wakeup. The Communication Controller provides to the Host the ability to transmit a special wakeup pattern on each of its available channels separately. The Communication Controller needs to recognize the wakeup pattern only during “WAKEUP” state.

Wakeup may be performed on only one channel at a time. The Host has to configure the wakeup channel while the Communication Controller is in “CONFIG” state by writing bit

FlexRay Protocol Controller (E-Ray)

SUCC1.WUCS in the SUC Configuration Register 1. The Communication Controller ensures that ongoing communication on this channel is not disturbed. The Communication Controller cannot guarantee that all nodes connected to the configured channel awake upon the transmission of the wakeup pattern, since these nodes cannot give feedback until the startup phase. The wakeup procedure enables single-channel devices in a two-channel system to trigger the wakeup, by only transmitting the wakeup pattern on the single channel to which they are connected. Any coldstart node that deems a system startup necessary will then wake the remaining channel before initiating communication startup.

The wakeup procedure tolerates any number of nodes simultaneously trying to wakeup a single channel and resolves this situation such that only one node transmits the pattern. Additionally the wakeup pattern is collision resilient, so even in the presence of a fault causing two nodes to simultaneously transmit a wakeup pattern, the resulting collided signal can still wake the other nodes.

After wakeup the Communication Controller returns to "READY" state and signals the change of the wakeup status to the Host by setting bit **SIR.WST** in the Status Service Request Register. The wakeup status vector can be read from the Communication Controller Status Vector register **CCSV.WSV**. If a valid wakeup pattern was received also either flag **SIR.WUPA** or flag **SIR.WUPB** in the Status Service Request Register is set.

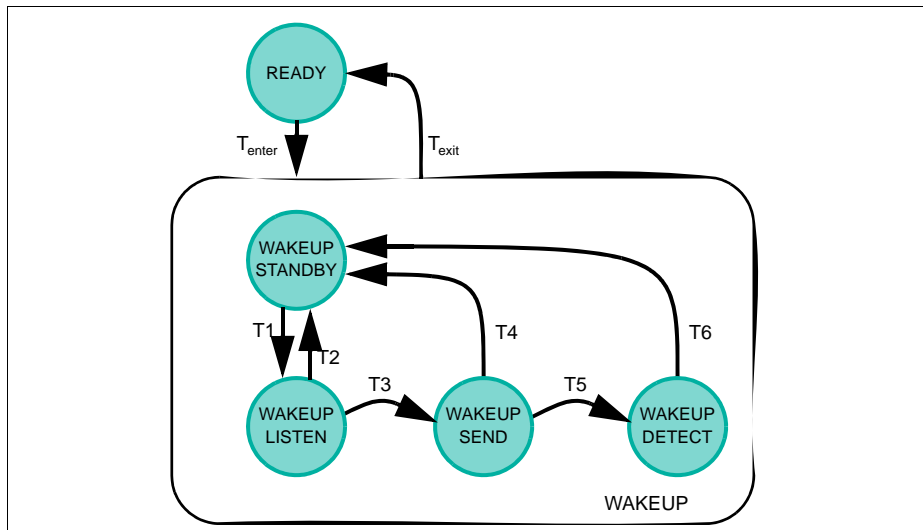


Figure 2-6 Structure of POC State WAKEUP

Table 2-12 State Transitions WAKEUP

T#	Condition	From	To
enter	Host commands change to "WAKEUP" state by writing SUCC1.CMD = 0011 _B (WAKEUP command)	READY	WAKEUP
1	CHI command WAKEUP triggers wakeup FSM to transit to "WAKEUP_LISTEN" state	WAKEUP_STANDBY	WAKEUP_LISTEN
2	Received WUP on wakeup channel selected by flag SUCC1.WUCS in the SUC Configuration Register 1 OR frame header on either available channel	WAKEUP_LISTEN	WAKEUP_STANDBY
3	Timer event	WAKEUP_LISTEN	WAKEUP_SEND
4	Complete, non-aborted transmission of wakeup pattern	WAKEUP_SEND	WAKEUP_STANDBY
5	Collision detected	WAKEUP_SEND	WAKEUP_DETECT
6	Wakeup timer expired OR WUP detected on wakeup channel selected by flag SUCC1.WUCS in the SUC Configuration Register 1 OR frame header received on either available channel	WAKEUP_DETECT	WAKEUP_STANDBY
exit	Wakeup completed (after T2 or T4 or T6) OR Host commands change to "READY" state by writing SUCC1.CMD = 0010 _B (READY command). This command also resets the wakeup FSM to "WAKEUP_STANDBY" state	WAKEUP	READY

The "WAKEUP_LISTEN" state is controlled by the wakeup timer and the wakeup noise timer. The two timers are controlled by the parameters listen timeout **SUCC2.LT** and listen timeout noise **SUCC2.LTN**. Both values can be configured in the SUC Configuration Register 2. listen timeout enables a fast cluster wakeup in case of a noise free environment, while listen timeout noise enables wakeup under more difficult conditions regarding noise interference.

In "WAKEUP_SEND" state the Communication Controller transmits the wakeup pattern on the configured channel and checks for collisions. After return from wakeup the Host has to bring the Communication Controller into "STARTUP" state by CHI command RUN.

In "WAKEUP_DETECT" state the Communication Controller attempts to identify the reason for the wakeup collision detected in "WAKEUP_SEND" state. The monitoring is

FlexRay Protocol Controller (E-Ray)

bounded by the expiration of listen timeout as configured by **SUCC2.LT** in the SUC Configuration Register 2. Either the detection of a wakeup pattern indicating a wakeup attempt by another node or the reception of a frame header indication existing communication, causes the direct transition to “READY” state. Otherwise WAKEUP_DETECT is left after expiration of listen timeout; in this case the reason for wakeup collision is unknown.

The Host has to be aware of possible failures of the wakeup and act accordingly. It is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal time it takes another coldstart node to become awake and to be configured.

The FlexRay Protocol Specification v2.1 recommends that two different Communication Controllers shall awake the two channels.

Host activities

The host must coordinate the wakeup of the two channels and must decide whether, or not, to wake a specific channel. The sending of the wakeup pattern is initiated by the Host and generated by the Communication Controller. The wakeup pattern is detected by the remote BDs and signalled to their local Hosts.

Wakeup procedure controlled by Host (single-channel wakeup):

- Configure the Communication Controller in “CONFIG” state
 - Select wakeup channel by programming bit **SUCC1.WUCS**
- Check local BDs whether a WUP was received
- Activate BD of selected wakeup channel
- Command Communication Controller to start wakeup on the configured channel by writing **SUCC1.CMD** = 0011_B
 - Communication Controller enters “WAKEUP”
 - Communication Controller returns to “READY” state and signals status of wakeup attempt to Host
- Wait predefined time to allow the other nodes to wakeup and configure themselves
- Coldstart node: wait for WUP on the other channel
 - In a dual channel cluster wait for WUP on the other channel
 - Reset coldstart inhibit flag **CCSV.CSI** by writing **SUCC1.CMD** = 1001_B (ALLOW_COLDSTART command)
 - Reset coldstart inhibit flag **CCSV.CSI** by writing **SUCC1.CMD** = 1001_B (ALLOW_COLDSTART command)
- Reset Coldstart Inhibit flag **CCSV.CSI** in the CCSV register by writing **SUCC1.CMD** = 1001_B (ALLOW_COLDSTART command), coldstart node only
- Command Communication Controller to enter startup by writing **SUCC1.CMD** = 0100_B (RUN command)

Wakeup procedure triggered by BD:

- Wakeup recognized by BD

FlexRay Protocol Controller (E-Ray)

- BD triggers power-up of Host (if required)
- BD signals wakeup event to Host
- Host configures its local Communication Controller
- If necessary Host commands wakeup of second channel and waits predefined time to allow the other nodes to wakeup and configure themselves
- Host commands Communication Controller to enter “STARTUP” state by writing **SUCC1.CMD** = 0100_B (RUN command)

Wakeup pattern (WUP)

The wakeup pattern is composed of at least two wakeup symbols (WUS). Wakeup symbol and wakeup pattern are configured by the PRT Configuration Registers **PRTC1** and **PRTC2**.

- Single channel wakeup, wakeup symbol may not be sent on both channels at the same time
- Wakeup symbol collision resilient for up to two sending nodes (two overlapping wakeup symbols still recognizable)
- Wakeup symbol must be configured identical in all nodes of a cluster
- Wakeup symbol transmit low time configured by **PRTC1.TXL**
- Wakeup symbol idle time used to listen for activity on the bus, configured by **PRTC1.TXI**
- A wakeup pattern composed of at least two Tx-wakeup symbols needed for wakeup
- Number of repetitions configurable by **PRTC1.RWP** (2 to 63 repetitions)
- Wakeup symbol receive window length configured by **PRTC1.RXW**
- Wakeup symbol receive low time configured by **PRTC1.RXL**
- Wakeup symbol receive idle time configured by **PRTC1.RXI**

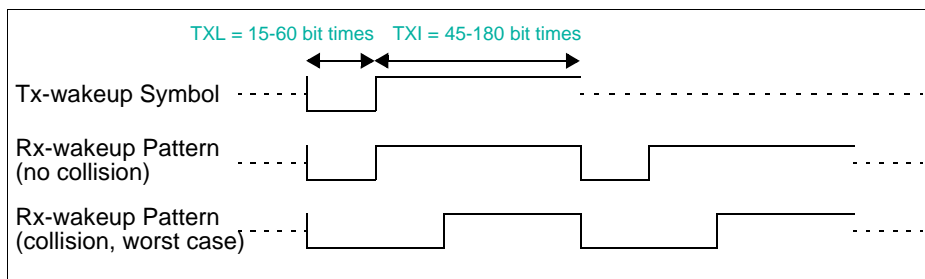


Figure 2-7 Timing of Wakeup Pattern

2.5.5.6 STARTUP State

The description below is intended to help configuring startup for the E-Ray IP-module. A detailed description of the startup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.2.

FlexRay Protocol Controller (E-Ray)

Any node entering “STARTUP” state that has coldstart capability should assure that both channels attached have been awakened before initiating coldstart.

It cannot be assumed that all nodes and stars need the same amount of time to become completely awake and to be configured. Since at least two nodes are necessary to start up the cluster communication, it is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal amount of time it takes another coldstart node to become awake, to be configured and to enter startup. It may require several hundred milliseconds (depending on the hardware used) before all nodes and stars are completely awakened and configured.

Startup is performed on all channels synchronously. During startup, a node only transmits startup frames.

A fault-tolerant, distributed startup strategy is specified for initial synchronization of all nodes. In general, a node may enter “NORMAL_ACTIVE” state via (see [Figure 2-8](#)):

- Coldstart path initiating the schedule synchronization (leading coldstart node)
- Coldstart path joining other coldstart nodes (following coldstart node)
- Integration path integrating into an existing communication schedule (all other nodes)

A coldstart attempt begins with the transmission of a collision avoidance symbol (CAS). Only a coldstart node that had transmitted the CAS transmits frames in the first four cycles after the CAS, it is then joined firstly by the other coldstart nodes and afterwards by all other nodes.

A coldstart node has the Transmit Sync Frame in Key Slot bits **SUCC1.TXST** and **SUCC1.TXSY** in the SUC Configuration Register 1 set to 1. The Message Buffer 0 holds the key slot ID which defines the slot number where the startup frame is sent. In the frame header of the startup frame the startup frame indicator bit is set.

In clusters consisting of three or more nodes, at least three nodes shall be configured to be coldstart nodes. In clusters consisting of two nodes, both nodes must be coldstart nodes. At least two fault-free coldstart nodes are necessary for the cluster to startup.

Each startup frame must also be a sync frame; therefore each coldstart node will also be a sync node. The number of coldstart attempts is configured by **SUCC1.CSA** in the SUC Configuration Register 1.

A non-coldstart node requires at least two startup frames from distinct nodes for integration. It may start integration before the coldstart nodes have finished their startup. It will not finish its startup until at least two coldstart nodes have finished their startup.

Both non-coldstart nodes and coldstart nodes start passive integration via the integration path as soon as they receive sync frames from which to derive the TDMA schedule information. During integration the node has to adapt its own clock to the global clock (rate and offset) and has to make its cycle time consistent with the global schedule observable at the network. Afterwards, these settings are checked for consistency with all available network nodes. The node can only leave the integration phase and actively participate in communication when these checks are passed.

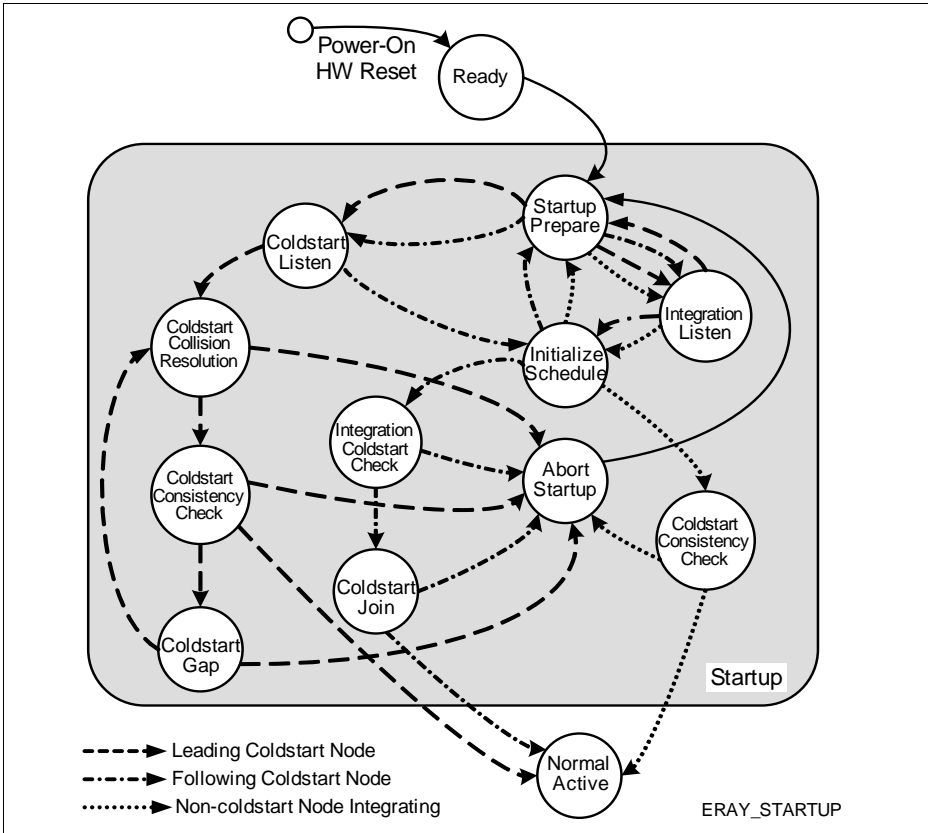


Figure 2-8 State Diagram Time-Triggered Startup

Coldstart Inhibit Mode

In coldstart inhibit mode the node is prevented from initializing the TDMA communication schedule. If bit **CCSV.CSI** in the Communication Controller Status Vector register is set, the node is not allowed to initialize the cluster communication, i.e. entering the coldstart path is prohibited. The node is allowed to integrate to a running cluster or to transmit startup frames after another coldstart node started the initialization of the cluster communication.

The coldstart inhibit bit **CCSV.CSI** is set whenever the POC enters "READY" state. The bit has to be cleared under control of the Host by CHI command **ALLOW_COLDSTART** (**SUCC1.CMD** = 1001_B)

2.5.5.7 Startup Timeouts

The Communication Controller supplies two different microtick timers supporting two timeout values, startup timeout and startup noise timeout. The two timers are reset when the Communication Controller enters the “COLDSTART_LISTEN” state. The expiration of either of these timers causes the node to leave the initial sensing phase (“COLDSTART_LISTEN” state) with the intention of starting up communication.

*Note: The startup and startup noise timers are identical with the wakeup and wakeup noise timers and use the same configuration values **SUCC2.LT** and **SUCC2.LTN** from the SUC Configuration Register 2.*

Startup Timeout

The startup timeout limits the listen time used by a node to determine if there is already communication between other nodes or at least one coldstart node actively requesting the integration of others.

The startup timer is configured by programming **SUCC2.LT** (pdListenTimeout) in the SUC Configuration Register 2.

The startup timer is restarted upon:

- Entering the “COLDSTART_LISTEN” state
- Both channels reaching idle state while in “COLDSTART_LISTEN” state

The startup timer is stopped:

- If communication channel activity is detected on one of the configured channels while the node is in the “COLDSTART_LISTEN” state
- When the “COLDSTART_LISTEN” state is left

Once the startup timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The timer status is kept for further processing by the startup state machine.

Startup Noise Timeout

At the same time the startup timer is started for the first time (transition from “STARTUP_PREPARE” state to “COLDSTART_LISTEN” state), the startup noise timer is started. This additional timeout is used to improve reliability of the startup procedure in the presence of noise.

The startup noise timer is configured by programming **SUCC2.LTN** (gListenNoise - 1) in the SUC Configuration Register 2 (see “**SUC Configuration Register 2 (SUCC2)**” on [Page 2-80](#)).

The startup noise timeout is:

$\text{pdListenTimeout} \cdot \text{gListenNoise} = \text{SUCC2.LT} \cdot (\text{SUCC2.LTN} + 1)$

The startup noise timer is restarted upon:

- Entering the “COLDSTART_LISTEN” state
- Reception of correctly decoded headers or CAS symbols while the node is in “COLDSTART_LISTEN” state

The startup noise timer is stopped when the “COLDSTART_LISTEN” state is left.

Once the startup noise timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The status is kept for further processing by the startup state machine. Since the startup noise timer won't be restarted when random channel activity is sensed, this timeout defines the fall-back solution that guarantees that a node will try to start up the communication cluster even in the presence of noise.

2.5.5.8 Path of leading Coldstart Node (initiating coldstart)

When a coldstart node enters “COLDSTART_LISTEN”, it listens to its attached channels.

If no communication is detected, the node enters the “COLDSTART_COLLISION_RESOLUTION” state and commences a coldstart attempt. The initial transmission of a CAS symbol is succeeded by the first regular cycle. This cycle has the number zero.

From cycle zero on, the node transmits its startup frame. Since each coldstart node is allowed to perform a coldstart attempt, it may occur that several nodes simultaneously transmit the CAS symbol and enter the coldstart path. This situation is resolved during the first four cycles after CAS transmission.

As soon as a node that initiates a coldstart attempt receives a CAS symbol or a frame header during these four cycles, it re-enters the “COLDSTART_LISTEN” state. Thereby, only one node remains in this path. In cycle four, other coldstart nodes begin to transmit their startup frames.

After four cycles in “COLDSTART_COLLISION_RESOLUTION” state, the node that initiated the coldstart enters the “COLDSTART_CONSISTENCY_CHECK” state. It collects all startup frames from cycle four and five and performs the clock correction. If the clock correction does not deliver any errors and it has received at least one valid startup frame pair, the node leaves “COLDSTART_CONSISTENCY_CHECK” and enters “NORMAL_ACTIVE” state.

The number of coldstart attempts that a node is allowed to perform is configured by **SUCC1.CSA** in the SUC Configuration Register 1. The number of remaining coldstarts attempts **CCSV.RCA** can be read from Communication Controller Status Vector register. The number of remaining attempts is reduced by one for each attempted coldstart. A node may enter the “COLDSTART_LISTEN” state only if this value is larger than one and it may enter the “COLDSTART_COLLISION_RESOLUTION” state only if this value is larger than zero. If the number of coldstart attempts is one, coldstart is inhibited but integration is still possible.

Path of following Coldstart Node (responding to leading Coldstart Node)

When a coldstart node enters the "COLDSTART_LISTEN" state, it tries to receive a valid pair of startup frames to derive its schedule and clock correction from the leading coldstart node.

As soon as a valid startup frame has been received the "INITIALIZE_SCHEDULE" state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and can derive a schedule from this startup frames, the "INTEGRATION_COLDSTART_CHECK" state is entered.

In "INTEGRATION_COLDSTART_CHECK" state it is assured that the clock correction can be performed correctly and that the coldstart node from which this node has initialized its schedule is still available. The node collects all sync frames and performs clock correction in the following double-cycle. If clock correction does not signal any errors and if the node continues to receive sufficient frames from the same node it has integrated on, the "COLDSTART_JOIN" state is entered.

In "COLDSTART_JOIN" state integrating coldstart nodes begin to transmit their own startup frames. Thereby the node that initiated the coldstart and the nodes joining it can check if their schedules agree to each other. If for the following three cycles the clock correction does not signal errors and at least one other coldstart node is visible, the node leaves "COLDSTART_JOIN" state and enters "NORMAL_ACTIVE" state. Thereby it leaves "STARTUP" at least one cycle after the node that initiated the coldstart.

Path of Non-coldstart Node

When a non-coldstart node enters the INTEGRATION_LISTEN state, it listens to its attached channels and tries to receive FlexRay frames.

As soon as a valid startup frame has been received the "INITIALIZE_SCHEDULE" state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_CONSISTENCY_CHECK state is entered.

In "INTEGRATION_CONSISTENCY_CHECK" state it is verified that the clock correction can be performed correctly and that enough coldstart nodes (at least 2) send startup frames that agree to the nodes own schedule. Clock correction is activated, and if any errors are signalled, the integration attempt is aborted.

During the first even cycle in this state, either two valid startup frames or the startup frame of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

During the first double-cycle in this state, either two valid startup frame pairs or the startup frame pair of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

FlexRay Protocol Controller (E-Ray)

If after the first double-cycle less than two valid startup frames are received within an even cycle, or less than two valid startup frame pairs are received within a double-cycle, the startup attempt is aborted.

Nodes in this state need to see two valid startup frame pairs for two consecutive double-cycles each to be allowed to leave STARTUP and enter NORMAL_OPERATION. Consequently, they leave startup at least one double-cycle after the node that initiated the coldstart and only at the end of a cycle with an odd cycle number.

2.5.5.9 NORMAL_ACTIVE State

As soon as the node that transmitted the first CAS symbol (resolving the potential access conflict and entering "STARTUP" via coldstart path) and one additional node have entered the "NORMAL_ACTIVE" state, the startup phase for the cluster has finished. In the "NORMAL_ACTIVE" state, all configured messages are scheduled for transmission. This includes all data frames as well as the sync frames. Rate and offset measurement is started in all even cycles (even/odd cycle pairs required).

In "NORMAL_ACTIVE" state the Communication Controller supports regular communication functions

- The Communication Controller performs transmissions and reception on the FlexRay bus as configured
- Clock synchronization is running
- The Host interface is operational

The Communication Controller exits from that state to

- "HALT" state by writing **SUCC1.CMD** = 0110_B (HALT command, at the end of the current cycle)
- "HALT" state by writing **SUCC1.CMD** = 0111_B (FREEZE command, immediately)
- "HALT" state due to change of the error state from "ACTIVE" to "COMM_HALT"
- "NORMAL_PASSIVE" state due to change of the error state from "ACTIVE" to "PASSIVE"
- "READY" state by writing **SUCC1.CMD** = 0010_B (READY command)

2.5.5.10 NORMAL_PASSIVE State

"NORMAL_PASSIVE" state is entered from "NORMAL_ACTIVE" state when the error state changes from ACTIVE (green) to PASSIVE (yellow).

In "NORMAL_PASSIVE" state, the node is able to receive all frames (node is fully synchronized and performs clock synchronization). In comparison to the "NORMAL_ACTIVE" state the node does not actively participate in communication, i.e. neither symbols nor frames are transmitted.

FlexRay Protocol Controller (E-Ray)

In "NORMAL_PASSIVE" state

- The Communication Controller performs reception on the FlexRay bus
- The Communication Controller does not transmit any frames or symbols on the FlexRay bus
- Clock synchronization is running
- The Host interface is operational

The Communication Controller exits from this state to

- "HALT" state by writing **SUCC1.CMD** = 0110_B (HALT command, at the end of the current cycle)
- "HALT" state by writing **SUCC1.CMD** = 0111_B (FREEZE command, immediately)
- "HALT" state due to change of the error state from "PASSIVE" to "COMM_HALT"
- "NORMAL_ACTIVE" state due to change of the error state from "PASSIVE" to "ACTIVE". The transition takes place when **CCEV.PTAC** from the Communication Controller Error Vector register equals **SUCC1.PTA** - 1.
- "READY" state by writing **SUCC1.CMD** = 0010_B (READY command)

2.5.5.11 HALT State

In this state all communication (reception and transmission) is stopped.

The Communication Controller enters this state

- By writing **SUCC1.CMD** = 0110_B (HALT command) while the Communication Controller is in "NORMAL_ACTIVE" or "NORMAL_PASSIVE" state
- By writing **SUCC1.CMD** = 0111_B (FREEZE command) from all states
- When exiting from "NORMAL_ACTIVE" state because the clock correction failed counter reached the "maximum without clock correction fatal" limit
- When exiting from "NORMAL_PASSIVE" state because the clock correction failed counter reached the "maximum without clock correction fatal" limit

The Communication Controller exits from this state to "CONFIG" state

- By writing **SUCC1.CMD** = 0001_B (DEFAULT_CONFIG command)

When the Communication Controller enters "HALT" state, all configuration and status data is maintained for analyzing purposes.

When the Host writes **SUCC1.CMD** = 0110_B (HALT command) in the SUC Configuration Register 1 to 1, the Communication Controller sets bit **CCSV.HRQ** in the Communication Controller Status Vector register and enters "HALT" state after the current communication cycle has finished.

When the Host writes **SUCC1.CMD** = 0111_B (FREEZE command) in the SUC Configuration Register to 1, the Communication Controller enters "HALT" state immediately and sets the **CCSV.FSI** bit in the Communication Controller Status Vector register.

FlexRay Protocol Controller (E-Ray)

The POC state from which the transition to HALT state took place can be read from **CCSV.PSL**.

2.5.6 Network Management

The accrued Network Management (NM) vector is located in the Network Management Register 1 to Network Management Register 3 (**NMVnn (nn = 1-3)**). The Communication Controller performs a logical OR operation over all network management (NM) vectors out of all received valid network management (NM) frames with the Payload Preamble Indicator (PPI) bit set. Only a static frame may be configured to hold network management (NM) information. The Communication Controller updates the network management (NM) vector at the end of each cycle.

The length of the network management (NM) vector can be configured from 0 to 12 byte by NML in the NEM Configuration Register. The network management (NM) vector length must be configured identically in all nodes of a cluster.

To configure a transmit buffer to send FlexRay frames with the PPI bit set, the PPIT bit in the header section of the respective transmit buffer has to be set via **WRHS1.PPIT**. In addition the Host has to write the network management (NM) information to the data section of the respective transmit buffer.

The evaluation of the network management (NM) vector has to be done by the application running on the Host.

*Note: In case a message buffer is configured for transmission / reception of network management frames, the payload length configured in header 2 of that message buffer should be equal or greater than the length of the NM vector configured by **NEMC.NML**.*

2.5.7 Filtering and Masking

Filtering is done by checking specific fields in a received frame against the corresponding configuration constants of the valid message buffers and the actual slot and cycle counter values (acceptance filtering), or by comparing the configuration constants of the valid message buffers against the actual slot and cycle counter values (transmit filtering). A message buffer is only updated / transmitted if the required matches occur.

Filtering is done on the following fields:

- Channel ID
- Frame ID
- Cycle Counter

The following filter combinations for acceptance / transmit filtering are allowed:

- Frame ID + Channel ID
- Frame ID + Channel ID + Cycle Counter

FlexRay Protocol Controller (E-Ray)

In order to store a received message in a message buffer all configured filters must match.

Note: For the FIFO the acceptance filter is configured by the FIFO Rejection Filter and the FIFO Rejection Filter Mask.

A message will be transmitted in the time slot corresponding to the configured frame ID on the configured channel(s). If cycle counter filtering is enabled the configured cycle filter value must also match.

2.5.7.1 Frame ID Filtering

Every transmit and receive buffer contains a frame ID stored in the header section. This frame ID is used differently for receive and transmit buffers.

Receive Buffers

A received message is stored in the first receive buffer where the received frame ID matches the configured frame ID, provided channel ID and cycle counter criteria are also met.

Transmit Buffers

For transmit buffers the configured frame ID is used to determine the appropriate slot for message transmission. The frame will be transmitted in the time slot corresponding to the configured frame ID, provided channel ID and cycle counter criteria are also met.

2.5.7.2 Channel ID Filtering

There is a 2-bit channel filtering field (CHA, CHB) located in the header section of each message buffer in the Message RAM. It serves as a filter for receive buffers, and as a control field for transmit buffers (see [Table 2-13](#)).

Table 2-13 Channel Filtering Configuration

CHA	CHB	Transmit Buffer transmit frame	Receive Buffer store valid receive frame
1	1	on both channels (static segment only)	received on channel A or B (store first semantically valid frame, static segment only)
1	0	on channel A	received on channel A
0	1	on channel B	received on channel B
0	0	no transmission	ignore frame

FlexRay Protocol Controller (E-Ray)

Note: If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted resp. received frames are ignored (same function as $CHA = CHB = 0$)

Receive Buffers

Valid received frames are stored if they are received on the channels specified in the channel filtering field. Only in static segment a receive buffer may be setup for reception on both channels (CHA and CHB set). Other filtering criteria must also be met.

If a valid header segment was stored, the respective MBC flag in the Message Buffer Status Changed register is set. If a valid payload segment was stored, the respective **NDn (n = 0-31) ... NDn (n = 96-127)** flag in the New Data **NDAT1 ... NDAT4** register is set. In both cases, if bit **RDHS1.MBI** in the header section of the respective message buffer is set, the RXI flag in the Status Service Request Register is set to 1. If enabled an service request is generated.

Transmit Buffers

The content of the buffer is transmitted only on the channels specified in the channel filtering field when the frame ID filtering and cycle counter filtering criteria are also met. Only in static segment a transmit buffer may be setup for transmission on both channels (CHA and CHB set). After transmission has completed, and if bit **WRHS1.MBI** in the header section of the respective message buffer is set, the TXI flag in the Status Service Request Register is set to 1. If enabled an service request is generated.

2.5.7.3 Cycle Counter Filtering

Cycle counter filtering is based on the notion of a cycle set. For filtering purposes, a match is detected if any one of the elements of the cycle set is matched. The cycle set is defined by the cycle code field in the header section of each message buffer.

If message buffer 0 is configured to hold the startup / sync frame or the single slot frame by bits TXST, TXSY, and TSM in the SUC Configuration Register 1, cycle counter filtering for message buffer 0 should be disabled.

*Note: Sharing of a static time slot via cycle counter filtering between different nodes of a FlexRay network is **not** allowed.*

The set of cycle numbers belonging to a cycle set is determined as described in [Table 2-14](#).

Table 2-14 Definition of Cycle Set

Cycle Code	Matching Cycle Counter Values		
000000 _B	all Cycles		
000001 _{C_B}	every second Cycle	at (Cycle Count)mod2	= c
00001 _{CC_B}	every fourth Cycle	at (Cycle Count)mod4	= cc
0001 _{CCC_B}	every eighth Cycle	at (Cycle Count)mod8	= ccc
001 _{CCCC_B}	every sixteenth Cycle	at (Cycle Count)mod16	= cccc
01 _{CCCCC_B}	every thirty-second Cycle	at (Cycle Count)mod32	= cccccc
1 _{CCCCCC_B}	every sixty-fourth Cycle	at (Cycle Count)mod64	= ccccccc

Table 2-15 below gives some examples for valid cycle sets to be used for cycle counter filtering:

Table 2-15 Examples for Valid Cycle Sets

Cycle Code	Matching Cycle Counter Values
0000011 _B	1-3-5-7--63 ↓
0000100 _B	0-4-8-12--60 ↓
0001110 _B	6-14-22-30--62 ↓
0011000 _B	8-24-40-56 ↓
0100011 _B	3-35 ↓
1001001 _B	9 ↓

Receive Buffers

The received message is stored only if the received cycle counter matches an element of the receive buffer's cycle set. Channel ID and frame ID criteria must also be met.

Transmit Buffers

The content of the buffer is transmitted on the configured channels when an element of the cycle set matches the current cycle counter value and the frame ID matches the slot counter value.

2.5.7.4 FIFO Filtering

For FIFO filtering there is one rejection filter and one rejection filter mask available. The FIFO rejection filter consists of 20 bits for **Channel** (2 bits), **Frame ID** (11 bits), and **Cycle Code** (7 bits). Rejection filter and rejection filter mask can be configured in

FlexRay Protocol Controller (E-Ray)

DEFAULT_CONGIF or "CONFIG" state only. The filter configuration in the header sections of message buffers belonging to the FIFO is ignored.

A valid received frame is stored in the FIFO if channel ID, frame ID, and cycle counter are not rejected by the configured rejection filter and rejection filter mask, and if there is no matching dedicated receive buffer.

2.5.8 Transmit Process

The transmit process is described in the following sections.

2.5.8.1 Static Segment

For the static segment, if there are several messages pending for transmission, the message with the frame ID corresponding to the next sending slot is selected for transmission.

The data section of transmit buffers assigned to the static segment can be updated until the end of the preceding time slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

2.5.8.2 Dynamic Segment

In the dynamic segment, if several messages are pending, the message with the highest priority (lowest frame ID) is selected next. Only frame ID's which are higher than the largest static frame ID are allowed for the dynamic segment.

In the dynamic segment different slot counter sequences are possible (concurrent sending of different frame ID's on both channels). Therefore pending messages are selected according to their frame ID and their channel configuration bit.

The data section of transmit buffers assigned to the dynamic segment can be updated until the end of the preceding slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

The start of latest transmit configured by SLT in the MHD Configuration Register 1 defines the maximum minislot value allowed before inhibiting new frame transmission in the dynamic segment of the current cycle.

2.5.8.3 Transmit Buffers

A portion of the E-Ray message buffers can be configured as transmit buffers by programming bit CFG in the header section of the respective message buffer to 1. This can be done via the Write Header Section 1 register.

FlexRay Protocol Controller (E-Ray)

There exist the following possibilities to assign a transmit buffer to the Communication Controller channels:

- Static segment: channel A **or** channel B, channel A **and** channel B
- Dynamic segment: channel A **or** channel B

Message buffer 0 is dedicated to hold the startup frame, the sync frame, or the designated single slot frame as configured by TXST, TXY, and TSM in the SUC Configuration Register 1. In this case it can be reconfigured in “DEFAULT_CONFIG” or “CONFIG” state only. This ensures that any node transmits at most one startup / sync frame per communication cycle. Transmission of startup / sync frames from other message buffers is not possible.

All other message buffers configured for transmission in static or dynamic segment are reconfigurable during runtime. Due to the organization of the data partition in the Message RAM (reference by data pointer), reconfiguration of the configured payload length and the data pointer in the header section of a message buffer may lead to erroneous configurations. If a message buffer is reconfigured during runtime it may happen that this message buffer is not send out in the respective communication cycle.

The Communication Controller does not have the capability to calculate the header CRC. The Host is supposed to provide the header CRCs for all transmit buffers. If network management is required the Host has to set the PPIT bit in the header section of the respective message buffer to 1 and write the network management information to the data section of the message buffer (see [Section 2.5.6](#)).

The payload length field configures the data payload length in 2-byte words. If the configured payload length of a static transmit buffer is shorter than the payload length configured for the static segment by SFDL in the Message Handler Configuration Register 1, the Communication Controller generates padding byte to ensure that frames have proper physical length. The padding pattern is logical zero.

Each transmit buffer provides a transmission mode flag TXM that allows the Host to configure the transmission mode for the transmit buffer in the static segment. If this bit is set, the transmitter operates in the single-shot mode. If this bit is cleared, the transmitter operates in the continuous mode. In dynamic segment the transmitter always works in single-shot mode.

If a message buffer is configured in the continuous mode, the Communication Controller does not reset the transmission request flag TXR after successful transmission. In this case a frame is sent out each time the frame ID and cycle counter filter match. The TXR flag can be reset by the Host by writing the respective message buffer number to the Input Buffer Command Request register while bit **STXRH** in the Input Buffer Command Mask register is reset to 0.

If two or more transmit buffers are configured with the same frame ID **and** cycle counter filter value, the transmit buffer with the lowest message buffer number will be transmitted in the respective slot.

2.5.8.4 Frame Transmission

To prepare a transmit buffer for transmission the following steps are required:

- Configure the message buffer as transmit buffer by writing bit CFG = 1 in the Write Header Section 1 register
- Write transmit message (header and data section) to the Input Buffer.
- To transfer a transmit message from Input Buffer to the Message RAM proceed as described on [“Data Transfer from Input Buffer to Message RAM” on Page 2-212](#).
- If configured in the Input Buffer Command Mask register the Transmission Request flag for the respective message buffer will be set as soon as the transfer has completed, and the message buffer is ready for transmission.
- Check whether the message buffer has been transmitted by checking the TXR bits (TXR = 0) in the Transmission Request 1,2 registers (single-shot mode only).

In single-shot mode the Communication Controller resets the TXR flag after transmission has been completed. Now the Host may update the transmit buffer with the next message. The Communication Controller does not transmit the message before the Host has indicated that the update is completed by setting the Transmission Request flag TXR again. The Host can check the actual state of the TXR flags of all message buffers by reading the Transmission Request registers. After successful transmission, if bit [WRHS1.MBI](#) in the header section of the respective message buffer is set, the transmit service request flag in the Status Service Request Register is set (TXI = 1). If enabled an service request is generated.

2.5.8.5 Null Frame Transmission

If in static segment the Host does not set the transmission request flag before transmit time, and if there is no other transmit buffer with matching frame ID and cycle counter filter, the Communication Controller transmits a null frame with the null frame indication bit set and the payload data reset to zero.

In the following cases the Communication Controller transmits a null frame with the null frame indication bit reset to 0, and the rest of the frame header and the frame length unchanged (payload data is reset to zero):

- All transmit buffers configured for the slot have cycle counter filters that do not match the current cycle
- There are matching frame ID's and cycle counter filters, but none of these transmit buffers has the transmission request flag TXR set

Null frames are not transmitted in the dynamic segment.

2.5.9 Receive Process

The receive process is described in the following sections.

2.5.9.1 Frame Reception

To prepare or change a message buffer for reception the following steps are required:

- Configure the message buffer as receive buffer by writing bit CFG = 0 in the Write Header Section 1 register
- Configure the receive buffer by writing the configuration data (header section) to the Input Buffer
- Transfer the configuration from Input Buffer to the Message RAM by writing the number of the target message buffer to the Input Buffer Command Request register.

Once these steps are performed, the message buffer functions as an active receive buffer and participates in the internal acceptance filtering process, which takes place every time the Communication Controller receives a message. The first matching receive buffer is updated from the received message. If the message buffer holds an unprocessed data section (ND = 1) it is overwritten with the new message and the MLST bit in the respective Message Buffer Status register is set.

If the payload length of a received frame PLC is longer than the value programmed by PLC in the header section of the respective message buffer, the data field stored in the message buffer is truncated to that length.

If no frame, a null frame, or a corrupted frame is received in a slot, the data section of the message buffer configured for this slot is not updated. In this case only the flags in the Message Buffer Status register are updated to signal the cause of the problem. In addition the respective MBC flag in the Message Buffer Status Changed 1,2,3,4 registers is set.

When the data section of a receive buffer has been updated from a received frame, the respective New Data **NDn (n = 0-31) ... NDn (n = 96-127)** flag in the New Data **NDAT1 ... NDAT4** registers is set. When the Message Handler has updated the message buffer status, the respective MBC flag in the Message Buffer Status Changed 1,2,3,4 registers is set. If bit **RDHS1.MBI** in the header section of the respective message buffer is set, the receive service request flag in the Status Service Request Register is set (RXI = 1). If enabled an service request is generated.

To read a receive buffer from the Message RAM via the Output Buffer proceed as described on **“Data Transfer from Message RAM to Output Buffer” on Page 2-215**.

Note: The ND and MBC flags are automatically cleared by the Message Handler when the received message has been transferred to the Output Buffer.

2.5.9.2 Null Frame reception

The payload segment of a received null frame is **not** copied into the matching receive buffer. If a null frame has been received, the header section of the matching message buffer is updated from the received null frame. The null frame indication bit in the header section 3 of the respective message buffer is reset (NFI = 0) and the respective MBC flag in the Message Buffer Status Changed 1,2,3,4 registers is set.

FlexRay Protocol Controller (E-Ray)

In case that bit ND and / or MBC were already set before this event because the Host did not read the last received message, bit MLST in the Message Buffer Status register of the respective message buffer is also set.

2.5.10 FIFO Function

A group of the message buffers can be configured as a cyclic First-In-First-Out (FIFO). The group of message buffers belonging to the FIFO is contiguous in the register map starting with the message buffer referenced by FFB and ending with the message buffer referenced by LCB in the Message RAM Configuration register. Up to 128 message buffers can be assigned to the FIFO.

2.5.10.1 Description

Every valid incoming message not matching with any dedicated receive buffer but passing the programmable FIFO filter is stored into the FIFO. In this case frame ID, payload length, receive cycle count, and the status bits of the addressed FIFO message buffer are overwritten with frame ID, payload length, receive cycle count, and the status from the received message and can be read by the Host for message identification. Bit RFNE in the Status Service Request Register shows that the FIFO is not empty, bit RFF in the Status Service Request Register is set when the last available message buffer belonging to the FIFO is written, bit RFO in the Error Service Request Register shows that a FIFO overrun has been detected. If enabled, service requests are generated.

There are two index registers associated with the FIFO. The PUT Index Register (PIDX) is an index to the next available location in the FIFO. When a new message has been received it is written into the message buffer addressed by the PIDX register. The PIDX register is then incremented and addresses the next available message buffer. If the PIDX register is incremented past the highest numbered message buffer of the FIFO, the PIDX register is loaded with the number of the first (lowest numbered) message buffer in the FIFO chain. The GET Index Register (GIDX) is used to address the next message buffer of the FIFO to be read. The GIDX register is incremented after transfer of the contents of a message buffer belonging to the FIFO to the Output Buffer. The PUT Index Register and the GET Index Register are not accessible by the Host.

The FIFO is completely filled when the PUT index (PIDX) reaches the value of the GET index (GIDX). When the next message is written to the FIFO before the oldest message has been read, both PUT index and GET index are incremented and the new message overwrites the oldest message in the FIFO. This will set FIFO overrun flag RFO in the Error Service Request Register.

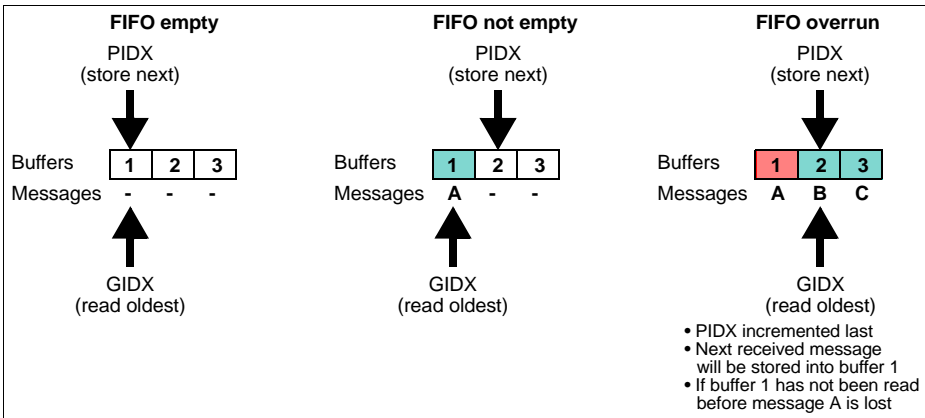


Figure 2-9 FIFO Status: Empty, Not Empty, Overrun

A FIFO non empty status is detected when the PUT index (PIDX) differs from the GET index (GIDX). In this case flag RFNE is set. This indicates that there is at least one received message in the FIFO. The FIFO empty, FIFO not empty, and the FIFO overrun states are explained in [Figure 2-9](#) for a three message buffer FIFO.

There is a programmable FIFO rejection filter for the FIFO. The FIFO Rejection Filter register (FRF) defines a filter pattern for messages to be rejected. The FIFO rejection filter consists of channel filter, frame ID filter, and cycle counter filter. If bit RSS is set to 1 (default), all messages received in the static segment are rejected by the FIFO. If bit RNF is set to 1 (default), received null frames are not stored in the FIFO.

The FIFO Rejection Filter Mask register (FRFM) specifies which bits of the frame ID filter in the FIFO Rejection Filter register are marked "don't care" for rejection filtering.

2.5.10.2 Configuration of the FIFO

For all message buffers belonging to the FIFO the data pointer to the first 32-bit word of the data section of the respective message buffer in the Message RAM has to be configured via the Write Header Section 3 register. All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask and needs not be configured in the header sections of the message buffers belonging to the FIFO.

When programming the data pointers for the message buffers belonging to the FIFO, the payload length of all message buffers should be programmed to the same value.

Note: It is recommended to program the MBI bits of the message buffers belonging to the FIFO to 0 via [WRHS1.MBI](#) to avoid generation of RX interrupts.

2.5.10.3 Access to the FIFO

To read from the FIFO the Host has to trigger a transfer from the Message RAM to the Output Buffer by writing the number of the first message buffer of the FIFO (referenced by FFB) to the Output Buffer Command Request register. The Message Handler then transfers the message buffer addressed by the GET Index Register (GIDX) to the Output Buffer. After this transfer the GET Index Register (GIDX) is incremented.

2.5.11 Message Handling

The Message Handler controls data transfers between the Input / Output Buffer and the Message RAM and between the Message RAM and the two Transient Buffer RAMs. All accesses to the internal RAM's are 32+1 bit accesses. The additional bit is used for parity checking.

Access to the message buffers stored in the Message RAM is done under control of the Message Handler state machine. This avoids conflicts between accesses of the two protocol controllers and the Host to the Message RAM.

Frame IDs of message buffers assigned to the static segment have to be in the range from 1 to NSS as configured in the GTU Configuration Register 7. Frame IDs of message buffers assigned to the dynamic segment have to be in the range from NSS + 1 to 2047.

Received messages with no matching dedicated receive buffer (static or dynamic segment) are stored in the receive FIFO (if configured) if they pass the FIFO rejection filter.

In RAM JTAG Enabled Mode the Message Handler is bypassed and all internal RAM blocks can directly be accessed via the Host interface (see [Section 2.4.2.2](#)).

2.5.11.1 Host access to Message RAM

The message transfer between Input Buffer and Message RAM as well as between Message RAM and Output Buffer is triggered by the Host by writing the number of the target / source message buffer to be accessed to the Input or Output Buffer Command Request register.

The Input / Output Buffer Command Mask registers can be used to write / read header and data section of the selected message buffer separately. If bit **STXRS** in the Input Buffer Command Mask register is set (**STXRS** = 1), the transmission request flag TXR of the selected message buffer is automatically set after the message buffer has been updated.

If bit **STXRS** in the Input Buffer Command Mask register is reset (**STXRS** = 0), the transmission request flag TXR of the selected message buffer is reset. This can be used to stop transmission from message buffers operated in continuous mode.

Input Buffer (IBF) and the Output Buffer (OBF) are build up as a double buffer structure. One half of this double buffer structure is accessible by the Host (IBF Host / OBF Host),

FlexRay Protocol Controller (E-Ray)

while the other half (IBF Shadow / OBF Shadow) is accessed by the Message Handler for data transfers between IBF / OBF and Message RAM.

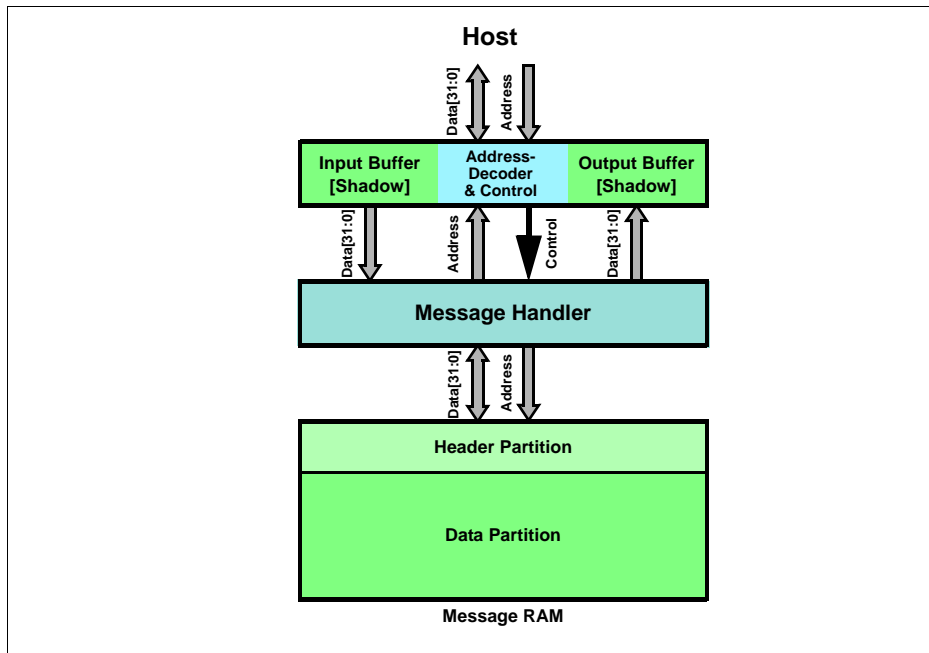


Figure 2-10 Host Access to Message RAM

Data Transfer from Input Buffer to Message RAM

To configure / update a message buffer in the Message RAM, the Host has to write the data to **WRDSnn (nn = 01-64)** and the header to **WRHS1**, **WRHS2**, **WRHS3**. The specific action is selected by configuring the Input Buffer Command Mask **IBCM**.

When the Host writes the number of the target message buffer in the Message RAM to **IBRH** in the Input Buffer Command Request register **IBCR**, IBF Host and IBF Shadow are swapped (see **Figure 2-11**).

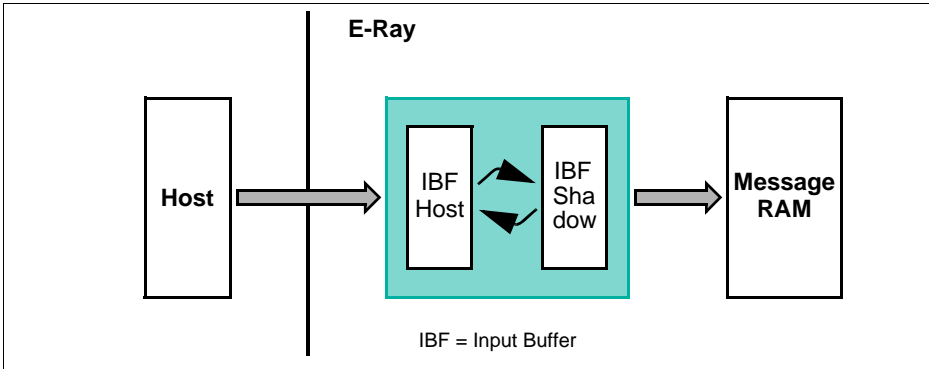


Figure 2-11 Double Buffer Structure Input Buffer

In addition the bits in the Input Buffer Command Mask and Input Buffer Command Request registers are also swapped to keep them attached to the respective IBF section (see [Figure 2-12](#)).

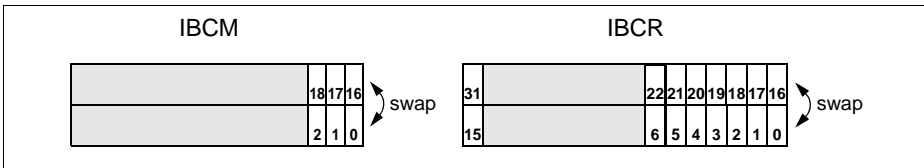


Figure 2-12 Swapping of IBCM and IBCR Bit

With this write operation the IBSYS bit in the Input Buffer Command Request register is set to 1. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by IBRS.

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message to IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, the IBSYS bit is set back to 0 and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to IBRH in the Input Buffer Command Request register.

If a write access to IBRH occurs while IBSYS is 1, IBSYH is set to 1. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, IBSYH is reset to 0, IBSYS remains set to 1, and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under IBRH and IBRS and the Command Mask flags are also swapped

Example of a 8/16/32-bit Host access sequence

Configure/update 1st message buffer via IBF

- Write data section to **WRDSnn (nn = 01-64)**
- Write header section to **WRHS1, WRHS2, WRHS3**
- Write Command Mask: write **IBCM.LHSH, IBCM.LDSH, IBCM.STXRH**
- Request data transfer to target message buffer: write IBRH
- Write data section to **WRDSnn (nn = 01-64)**
- Write header section to **WRHS1, WRHS2, WRHS3**
- Write Command Mask: write **IBCM.LHSH, IBCM.LDSH, IBCM.STXRH**
- Request data transfer to target message buffer: write IBRH after IBSYH is reset

Configure/update 3rd message buffer via IBF

- ...

Table 2-16 Assignment of Input Buffer Command Mask Bit

Pos.	Access	Bit	Function
18	rh	STXRS	Set Transmission Request Shadow
17	rh	LDSS	Load Data Section Shadow
16	rh	LHSS	Load Header Section Shadow
2	rw	STXRH	Set Transmission Request Host
1	rw	LDSH	Load Data Section Host
0	rw	LHSH	Load Header Section Host

Table 2-17 Assignment of Input Buffer Command Request Bit

Pos.	Access	Bit	Function
31	rh	IBSYS	IBF Busy Shadow, signals ongoing transfer from IBF Shadow to Message RAM
21...16	rh	IBRS	IBF Request Shadow, number of message buffer currently / last updated
15	rh	IBSYH	IBF Busy Host, transfer request pending for message buffer referenced by IBRH
5...0	rwh	IBRH	IBF Request Host, number of message buffer to be updated next

FlexRay Protocol Controller (E-Ray)

Data Transfer from Message RAM to Output Buffer

To read a message buffer from the Message RAM, the Host has to write to Command Request register **OBCR** to trigger the data transfer as configured in Output Buffer Command Mask **OBCM** register. After the transfer has completed, the Host can read the transferred data from **RDDSnn (nn = 01-64)**, **RDHS1**, **RDHS2**, **RDHS2**, and **MBS**.

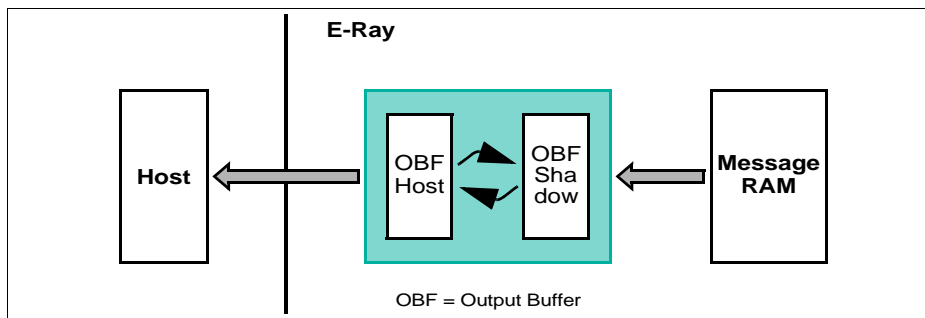


Figure 2-13 Double Buffer Structure Output Buffer

OBF Host and OBF Shadow as well as bits **OBCM.RHSS**, **OBCM.RDSS**, **OBCM.RHSH**, **OBCM.RDSH** and bits **OBCR.OBRS**, **OBCR.OBRH** are swapped under control of bits **OBCR.VIEW** and **OBCR.REQ**.

Writing bit **OBCR.REQ** to 1 copies bits **OBCM.RHSS**, **OBCM.RDSS** and bits **OBCR.OBRS** to an internal storage (see [Figure 2-14](#)).

After setting **OBCR.REQ** to 1, **OBCR.OBSYS** is set to 1, and the transfer of the message buffer selected by **OBCR.OBRS** from the Message RAM to OBF Shadow is started. After the transfer between the Message RAM and OBF Shadow has completed, the **OBCR.OBSYS** bit is set back to 0. Bits **OBCR.REQ** and **OBCR.VIEW** can only be set to 1 while **OBCR.OBSYS** is 0.

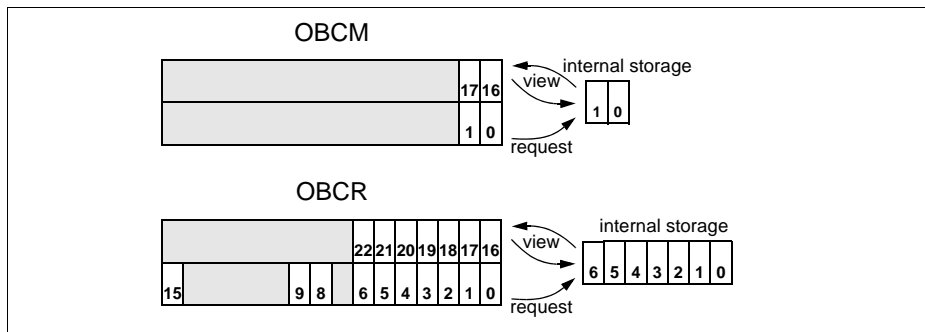


Figure 2-14 Swapping of OBCM and OBCR Bit

FlexRay Protocol Controller (E-Ray)

OBF Host and OBF Shadow are swapped by setting bit **OBCR.VIEW** to 1 while bit **OBCR.OBSYS** is 0 (see **Figure 2-13**).

In addition bits **OBCR.OBRH** and bits **OBCM.RHSH**, **OBCM.RDSH** are swapped with the registers internal storage thus assuring that the message buffer number stored in **OBCR.OBRH** and the mask configuration stored in **OBCM.RHSH**, **OBCM.RDSH** matches the transferred data stored in OBF Host (see **Figure 2-14**).

Now the Host can read the transferred message buffer from OBF Host while the Message Handler may transfer the next message from the Message RAM to OBF Shadow.

Example of an 8/16/32-bit Host access sequence:

Request transfer of 1st message buffer to OBF Shadow

- Wait until **OBCR.OBSYS** is reset
- Write Output Buffer Command Mask **OBCM.RHSS**, **OBCM.RDSS** for 1st message buffer
- Request transfer of 1st message buffer to OBF Shadow by writing **OBCR.OBRS** and **OBCR.REQ** (in case of an 8-bit Host interface, **OBCR.OBRS** has to be written before **OBCR.REQ**).
- Read out 1st transferred message buffer by reading **RDDSnn (nn = 01-64)**, **RDHS1**, **RDHS2**, **RDHS2**, and **MBS**

...

FlexRay Protocol Controller (E-Ray)

Demand access to last requested message buffer without request of another message buffer:

- Wait until **OBCR.OBSYS** is reset
- Demand access to last transferred message buffer by writing **OBCR.VIEW**
- Read out last transferred message buffer by reading **RDDSn** (**nn = 01-64**), **RDHS1**, **RDHS2**, **RDHS2**, and **MBS**

Table 2-18 Assignment of Output Buffer Command Mask Bit

Pos.	Access	Bit	Function
17	rh	RDSH	Data Section available for Host access
16	rh	RHSH	Header Section available for Host access
1	rw	RDSS	Read Data Section Shadow
0	rw	RHSS	Read Header Section Shadow

Table 2-19 Assignment of Output Buffer Command Request Bit

Pos.	Access	Bit	Function
22...16	rh	OBRH	OBF Request Host , number of message buffer available for Host access
15	rh	OBSYS	OBF Busy Shadow , signals ongoing transfer from Message RAM to OBF Shadow
9	rw	REQ	Request Transfer from Message RAM to OBF Shadow
8	rwh	VIEW	View OBF Shadow, swap OBF Shadow, and OBF Host
6...0	rwh	OBR	OBF Request Shadow , number of message buffer for next request

2.5.11.2 Data Transfers between IBF / OBF and Message RAM

This document uses the following terms and abbreviations:

Table 2-20 Terms and Abbreviations

Term	Meaning
MHD	Message Handler
IBF	Input Buffer 1 or 2 RAM
OBF	Output Buffer 1 or 2 RAM
MBF	Message Buffer RAM

Table 2-20 Terms and Abbreviations (cont'd)

TBF	Transient Buffer RAM Channel A (TBF1) or Channel B (TBF2)
IBF \Rightarrow MBF	Transfer from IBF to MBF
MBF \Rightarrow OBF	Transfer from MBF to OBF
MBF \Rightarrow TBF	Transfer from MBF to TBF
TBF \Rightarrow MBF	Transfer from TBF to MBF
SS	Slot Status
SS \Rightarrow MBF	Transfer SS to MBF

Message Handler functionality

The MHD controls the access to the MBF. It manages data-transfer between MBF and IBF, OBF, TBF1, TBF2. The data-path are shown in Figure 2-15.

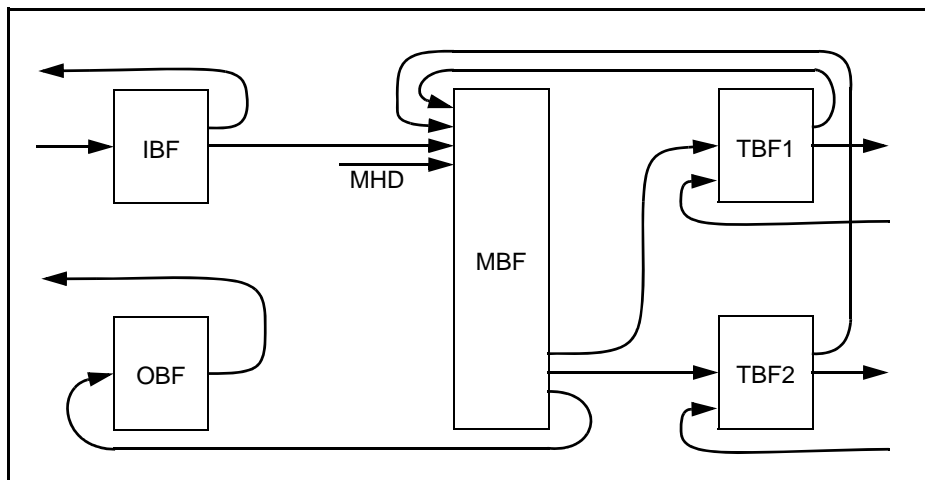


Figure 2-15 Interconnection of RAMs

Furthermore a search-algorithm allows to find the next valid message object in the MBF for transmission or reception.

Each transfer consists of a setup-time, four time steps to transfer the header-section and a payload-length-dependent number of time steps to transfer the data-section. The internal data-busses have a width of 32 bits. Thereby it is possible to transfer two 2-byte words in one time step. If the payload consists of an odd number of 2-byte words the last time step of the data-section contains only 16 bit of valid data. If the Payload-Length (PL) is e.g. 7, the data-section consists of 4 time steps.

FlexRay Protocol Controller (E-Ray)

The maximum length for the data-section is 64 time steps, the minimum length is zero time steps.

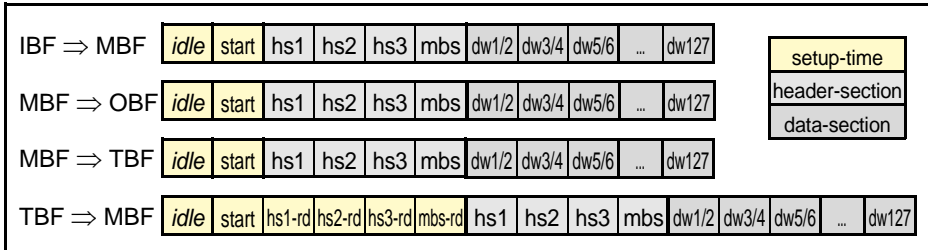


Figure 2-16 Different Possible Buffer Transfers

The update of the Slot-Status consists of a setup-time and one time-step to write the new Slot-Status.

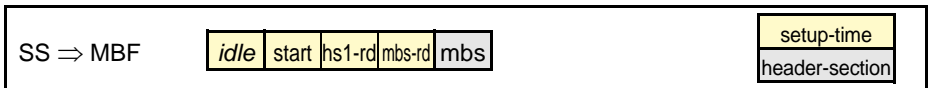


Figure 2-17 Update of Slot Status

The length of a time step depends on the number of concurrent tasks.

The following concurrent tasks are executed under control of the Message Handler:

- Data transfer between IBF or OBF and MBF
- Data transfer between TBF1 and MBF, search next TX / RX message buffer CHA
- Data transfer between TBF2 and MBF, search next TX / RX message buffer CHB

Thereby the time step length can vary between one and three eray_bclk periods.

Under certain conditions it is possible that a transfer is stopped or interrupted for a number of time steps until it is continued.

When a IBF \Rightarrow MBF is started short after a TBF \Rightarrow MBF or SS \Rightarrow MBF the transfer from IBF has to wait until the setup-time of the internal transfer has finished (see Figure 2-18)

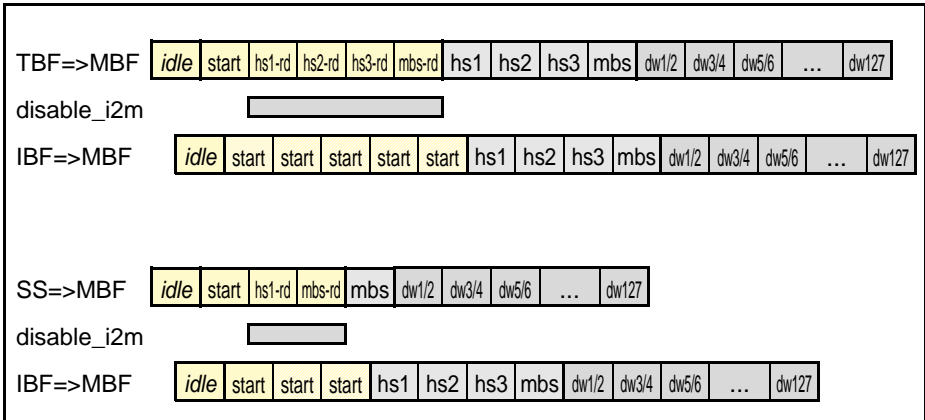


Figure 2-18 Delay start of IBF=>MBF

The internal signal “disable_i2m” is always active when the TBF \Rightarrow MBF is in state “hs1-rd”, “hs2-rd”, “hs3-rd” or “mbs-rd” and when the SS \Rightarrow MBF is in state “hs1-rd” or “mbs-rd”.

The IBF \Rightarrow MBF is hold in state “start” until the internal signal “disable_i2m” gets inactive.

These additional time-steps are independent of any address-counter-values. This means, the IBF \Rightarrow MBF has to wait even if it writes to another buffer than the internal transfer.

Multiple requests of transfers between IBF/OBF and Message RAM

The time required to transfer the contents of a message buffer between IBF / OBF and Message RAM depends on the number of 4-byte words to be transferred, the number of concurrent tasks to be managed by the Message Handler, and in special cases the type and address range of the internal transfer. The number of 4-byte words varies from 4 (header section only) to 68 (header + maximum data section) plus a short setup time to start the first transfer, while the number of concurrent task varies from one to three. The 4 header words have to be included in calculation even if only the data section is requested for transfer.

The following concurrent tasks are executed under control of the Message Handler:

- Data transfer between IBF or OBF and MBF
- Data transfer between TBF1 and MBF, search next TX / RX message buffer CHA
- Data transfer between TBF2 and MBF, search next TX / RX message buffer CHB

Transfers between IBF and MBF respectively MBF and OBF can only be handled one after another. In case that e.g. a IBF \Rightarrow MBF has been started shortly before a

FlexRay Protocol Controller (E-Ray)

MBF \Rightarrow OBF is requested, the MBF \Rightarrow OBF has to wait until the IBF \Rightarrow MBF has completed.

In case that e.g. a second IBF \Rightarrow MBF is requested, a MBF \Rightarrow OBF is requested and a IBF \Rightarrow MBF is ongoing, the MBF \Rightarrow OBF has to wait until the first IBF \Rightarrow MBF has completed. The second IBF \Rightarrow MBF has to wait until the MBF \Rightarrow OBF has completed (see figure 2-19) independent whether MBF \Rightarrow OBF or second IBF \Rightarrow MBF is requested first.

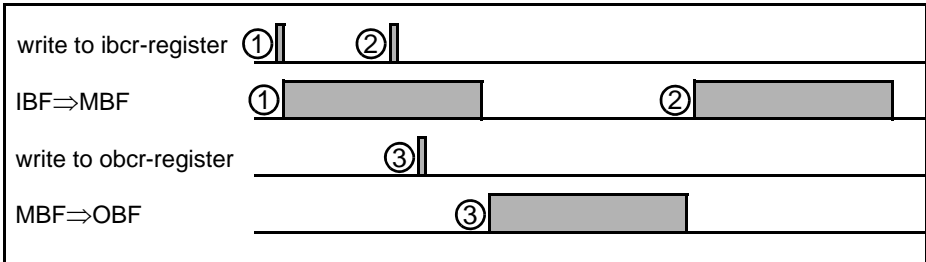


Figure 2-19 Multiple IBF/OBF Request

Worst case for single request

When a message with a large payload length is received the TBF \Rightarrow MBF is started at the begin of the next slot (n+1). If the next slot is a dynamic slot without transmission/reception (minislot), it may happen that the TBF \Rightarrow MBF has not finished until begin of the next but one slot (n+2). In this case the TBF \Rightarrow MBF will be service requested (break) to start a transmission in the next but one slot (MBF \Rightarrow TBF) and/or to update the slot status (SS \Rightarrow MBF) for the RX-buffer corresponding with next slot (n+1). After this interruption the TBF \Rightarrow MBF is continued.

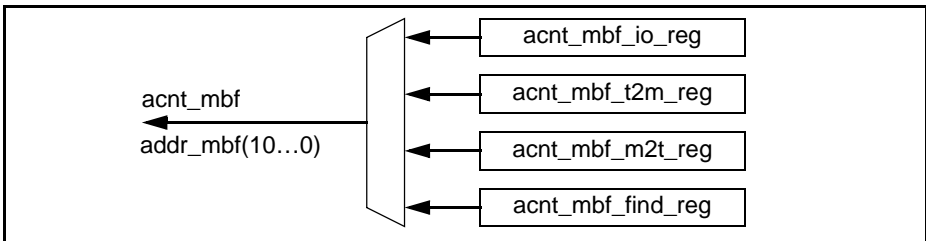


Figure 2-20 Address Counter Scheme of Message RAM (simplified)

For the transfers IBF \Rightarrow MBF / MBF \Rightarrow OBF, TBF \Rightarrow MBF and MBF \Rightarrow TBF separate address-counter are implemented (see Figure 2-20).

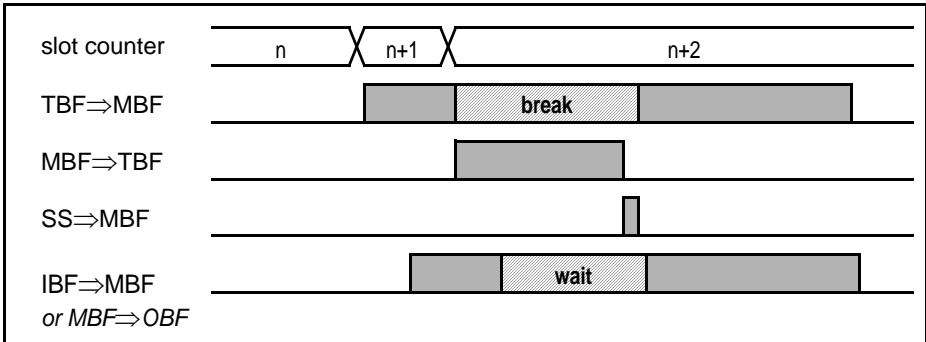


Figure 2-21 interruption of TBF=>MBF

If the address-counter for IBF=>MBF / MBF=>OBF (acnt_mbf_io_reg) reaches the address of the interrupted TBF=>MBF (acnt_mbf_t2m_reg) the IBF=>MBF / MBF=>OBF has to wait until the TBF=>MBF is continued (see Figure 2-21).

The relative time is measured in eray_bclk cycles. Absolute time depends on the actual eray_bclk cycle period.

$$\begin{aligned}
 \text{tbf_to_mbf_break time}_{\text{max}} &= (\text{setup time} + \text{mbf_to_tbf time}_{\text{max}}) \\
 &\quad + (\text{setup time} + \text{ss_to_mbf}) \\
 \text{cycles}_{\text{req}} &= (\text{number of concurrent tasks}) \\
 &\quad \times ((\text{setup time} + (\text{number of 4-byte words})_{\text{req}}) \\
 &\quad \quad + \text{tbf_to_mbf_break time}) \\
 \text{setup time} &= 2 \text{ eray_bclk cycles}
 \end{aligned}$$

FlexRay Protocol Controller (E-Ray)

Worst case for one IBF⇒MBF or MBF⇒OBF:

$$\text{Max. break time: } \text{tbf_to_mbf_break time}_{\text{max}} = (2+68) + (4+1) = 75$$

$$\text{Max. number of eray_bclk cycles: } \text{cycles}_{\text{req}} = 3 \times (6 + 68 + 75) = 435$$

Worst case for multiple transfers

If a second IBF⇒MBF and a MBF⇒OBF (see Figure 2-19) is requested directly after the first IBF⇒MBF has started following worst case timing could appear:

$$\begin{aligned} \text{cycles}_{\text{trans}} = & \text{ (remaining cycles of transfer running)} \\ & + \text{ (cycles of second requested transfer)} \\ & + \text{ (cycles of third requested transfer)} \end{aligned}$$

$$\text{cycles}_{\text{trans}} = \text{cycles}_{\text{rem}} + \text{cycles}_{\text{req_2}} + \text{cycles}_{\text{req_3}}$$

$$\text{Max. number of eray_bclk cycles: } \text{cycles}_{\text{trans}} = 447 + 435 + 447 = 1329$$

2.5.11.3 Minimum eray_bclk

To calculate the minimum eray_bclk the worst case scenario has to be considered.

The worst case scenario depends on the following parameters

- maximum payload length
- minimum minislot length
- number of configured message buffers (excluding FIFO)
- used channels (single/dual channel)

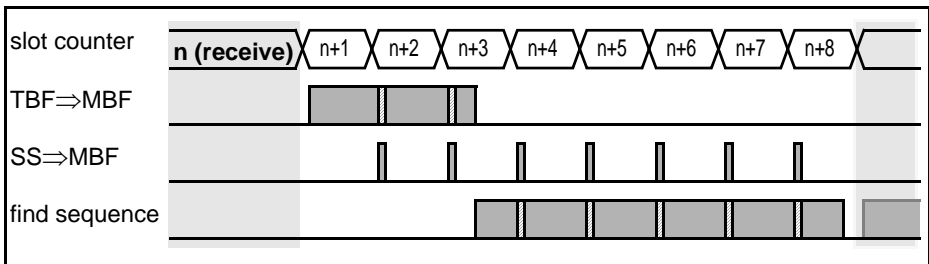


Figure 2-22 worst case scenario

FlexRay Protocol Controller (E-Ray)

Worst case scenario:

- reception of message with a maximum payload length in Slot n (n is 7,15,23,31,39,...)
- slot n+1 to n+7 are empty dynamic slots (minislot) and configured as receive buffer
- the find-sequence (usually started in slot 8,16,24,32,40,...) has to scan the maximum number of configured buffers
- the number of concurrent tasks has its maximum value of three

The find-sequence is executed each 8 Slots (slot 8,16,24,32,40,...). It has to be finished until the next find-sequence is requested.

The length of a TBF⇒MBF varies from 4 (header section only) to 68 (header + maximum data section) time step plus a setup time of 6 time steps.

$$\text{blk cycles}_{t_{2m}} = (\text{number of concurrent tasks}) \times (\text{setup time}_{t_{2m}} + (\text{number of 4-byte words})_{t_{2m}})$$

A SS⇒MBF has a fixed length of 1 time steps plus a setup time of 4 time steps.

$$\text{blk cycles}_{ss_{2m}} = (\text{number of concurrent tasks}) \times 5$$

The find sequence has a maximum length of 128 (maximum number of buffers) time steps plus a setup time of 2 time steps.

$$\text{blk cycles}_{\text{find}} = (\text{number of concurrent tasks}) \times (\text{setup time}_{\text{find}} + (\text{number of configured buffers}))$$

A minislot has a length of 2 to 63 macrotick (gdMinislot). The minimum nominal macrotick period (cdMinMTNom) is 1 μs. A sequence of 8 minislots has a length of

$$\text{time}_{8\text{minislots}} = 8 \times \text{gdMinislot} \times \text{cdMinMTNom}$$

FlexRay Protocol Controller (E-Ray)

The maximum eray_bclk period can be calculated as followed:

$$\text{time}_{8\text{minislots}} \geq (\text{bclk period in } \mu\text{s}) \times ((\text{bclk cycles}_{t2m}) + 7 \times (\text{bclk cycles}_{ss2m}) + (\text{bclk cycles}_{find}))$$

$$\text{bclk period in ms} \leq \frac{\text{time}_{8\text{minislots}}}{(\text{bclk cycles}_{t2m}) + 7 \times (\text{bclk cycles}_{ss2m}) + (\text{bclk cycles}_{find})}$$

$$\text{minimum time}_{8\text{minislots}} = 8 \times 2 \times 1 \mu\text{s} = 16 \mu\text{s}$$

$$\text{maximum bclk cycles}_{t2m} = 3 \times (6 + 68) = 222$$

$$\text{maximum bclk cycles}_{ss2m} = 3 \times 5 = 15$$

$$\text{maximum bclk cycles}_{find} = 3 \times (2 + 128) = 390$$

$$\text{eray_bclk period in ms} \leq \frac{16 \mu\text{s}}{222 + 7 \times 15 + 390} = 22.315 \dots \text{ns}$$

The minimum eray_bclk frequency for this worst case scenario is 44.8125 MHz.

A too low eray_bclk frequency can cause a malfunction of the E-Ray.

The E-Ray can detect several malfunctions and reports this by setting the corresponding flag in the Message Handler Constraints Flags (**MHDF**) register.

Minimum eray_bclk for various maximum payload length

Table 2-21 summarizes the minimum required eray_bclk frequency for various maximum payload length assuming:

- a minimum minislot length of 2 μs .
- a maximum of 128 configured message buffers.
- dual channels in use.

Table 2-21 Minimum eray_bclk for different maximum payload length

Maximum payload length of 32 bit words	4	8	16	32	64
minimum eray_bclk	32,82 MHz	33,57 MHz	35,07 MHz	38,07 MHz	44,82 MHz

Minimum eray_bclk for various minimum minislot length

Table 2-22 summarizes the minimum required eray_bclk frequency for various minimum minislot length assuming:

- a maximum payload length of 254 bytes / 64 four-byte-words.
- a maximum 128 configured message buffers.
- dual channels in use.

Table 2-22 Minimum eray_bclk for different minimum minislot length

gdMinislot at dMinMTNom = 1 μ s	2 μ s	3 μ s	4 μ s	7 μ s	8 μ s
minimum eray_bclk	44,82 MHz	29,88 MHz	22,412 MHz	12,8 MHz	9,96 MHz

Minimum eray_bclk for various amount of configured message buffers

Table 2-23 summarizes the minimum required eray_bclk frequency for various amount of configured Message Buffers assuming:

- a maximum payload length of 254 bytes / 64 four-byte-words.
- a minimum minislot length of 2 μ s.
- dual channels in use.

Table 2-23 Minimum eray_bclk for different amount of configured message buffers

Configured maximum amount of Message Buffers	128	64	32
minimum eray_bclk	44,82 MHz	32,82 MHz	26,82 MHz

Minimum eray_bclk for a typical configuration

‘The minimum required eray_bclk frequency for various assuming the following typical E-Ray configuration:

- a maximum payload length of 32 bytes / 8 four-byte-words.
- a minimum minislot length of 7 μ s.
- a maximum 128 configured message buffers.
- dual channels in use

The minimum eray_bclk frequency for this ‘typical example would be 10 MHz.

2.5.11.4 FlexRay Protocol Controller access to Message RAM

The two Transient Buffer RAMs (TBF 1, TBF 2) are used to buffer the data for transfer between the two FlexRay Protocol Controllers and the Message RAM.

Each Transient Buffer RAM is build up as a double buffer, able to store two complete FlexRay messages. There is always one buffer assigned to the corresponding Protocol Controller while the other one is accessible by the Message Handler.

If e.g. the Message Handler writes the next message to be send to Transient Buffer Tx, the FlexRay Channel Protocol Controller can access Transient Buffer Rx to store the message it is actually receiving. During transmission of the message stored in Transient Buffer Tx, the Message Handler transfers the last received message stored in Transient Buffer Rx to the Message RAM (if it passes acceptance filtering) and updates the respective message buffer.

Data transfers between the Transient Buffer RAMs and the shift registers of the FlexRay Channel Protocol Controllers are done in words of 32 bit. This enables the use of a 32 bit shift register independent of the length of the FlexRay messages.

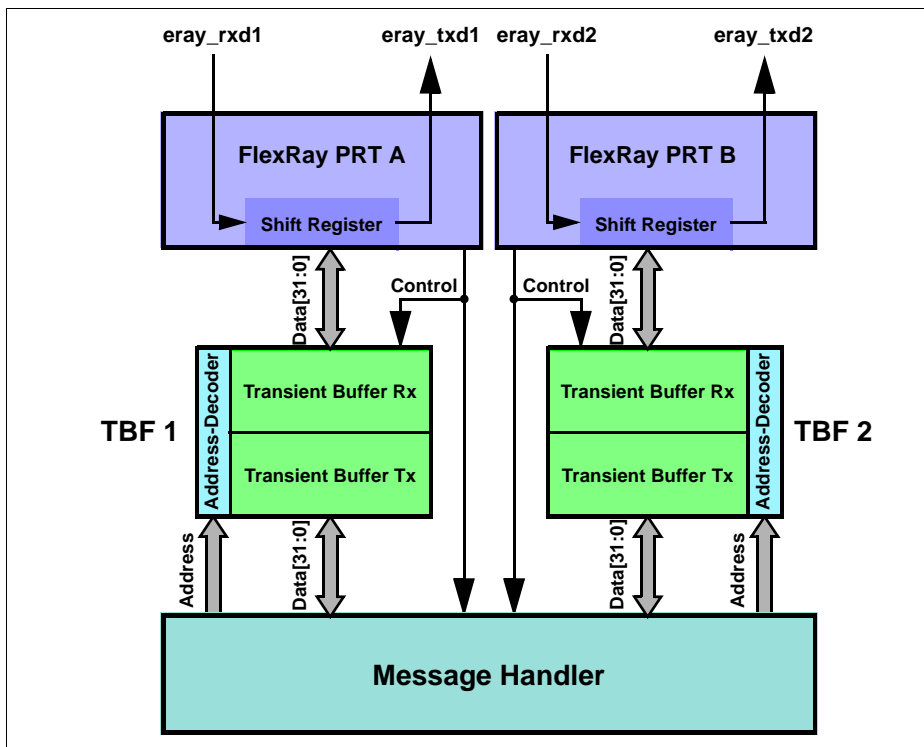


Figure 2-23 Access to Transient Buffer RAMs

2.5.12 Message RAM

To avoid conflicts between Host access to the Message RAM and FlexRay message reception / transmission, the Host cannot directly access the message buffers in the Message RAM. These accesses are handled via the Input and Output Buffers. The Message RAM is able to store up to 128 message buffers depending on the configured payload length.

The Message RAM is organized $2048 \times 33 = 67,584$ bit. Each 32-bit word is protected by a parity bit. To achieve the required flexibility with respect to different numbers of data byte per FlexRay frame (0...254), the Message RAM has a structure as shown in [Figure 2-24](#).

The data partition is allowed to start at Message RAM word number: $(MRC.LCB + 1) \cdot 4$

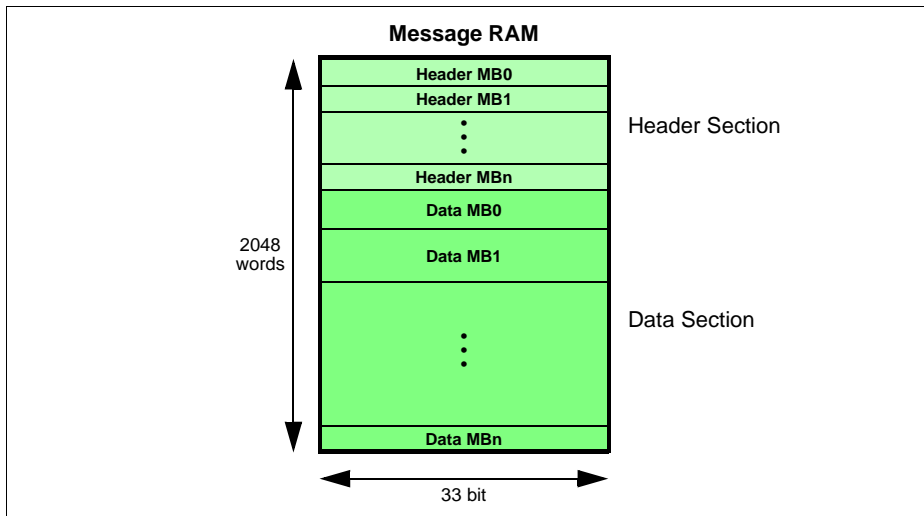


Figure 2-24 Structure of Message RAM

Header Partition

Stores header segments of FlexRay frames:

- Supports a maximum of 128 message buffers
- Each message buffer has a header of four 32+1 bit words
- Header 3 of each message buffer holds the 11 bit pointer to respective data section in the data partition

Data Partition

Flexible storage of data sections with different length. Some maximum values are:

- 30 message buffers with 254 byte data section each
- Or 56 message buffers with 128 byte data section each
- Or 128 message buffers with 48 byte data section each

Restriction: header partition + data partition may not occupy more than 2048 33-bit words.

2.5.12.1 Header Partition

The elements used for configuration of a message buffer as well as the actual message buffer status are stored in the header partition of the Message RAM as listed in [Table 2-24](#) below. Configuration of the header sections of the message buffers is done via IBF (Write Header Section 1...3). Read access to the header section is done via OBF (Read Header Section 1...3 + Message Buffer Status). The Data Pointer has to be calculated by the programmer to define the starting point of the data section for the respective message buffer in the data partition of the Message RAM. The Data Pointer should not be modified during runtime. For message buffers belonging to the receive FIFO (re)configuration should be done in "DEFAULT_CONFIG" or "CONFIG" state only.

The header of each message buffer occupies four 33-bit words in the header partition of the Message RAM. The header of message buffer 0 starts with the first word in the Message RAM.

For transmit buffers the Header CRC has to be calculated by the Host.

Payload Length Received **PLR**, Receive Cycle Count **RCC**, Received on Channel Indication **RCI**, Startup Frame Indication bit **SFI**, Sync bit **SYN**, Null Frame Indication bit **NFI**, Payload Preamble Indication bit **PPI**, and Reserved bit **RES** are only updated from received valid frames (including valid null frames).

Header word 4 of each configured message buffer holds the respective Message Buffer Status **MBS** information.

Table 2-24 Header Section of a Message Buffer in the Message RAM

Bit Word	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	P			M B I	T X I	N M E	C F G	C H B	C H A	Cycle Code												Frame ID											
2	P		Payload Length Received						Payload Length Configured										Tx Buffer: Header CRC Configured Rx Buffer: Header CRC Received														
3	P			R E S	P E S	N I N	S I N	S I N	R E C		Receive Cycle Count										Data Pointer												
4	P			R E S	P E S	N I N	S I N	S I N	R E C		Cycle Count Status						T F B	F T Y			M E S	E L S	T S C	T C C	S V V	S C C	S E E	S E E	S E E	V E E	V E E	V E E	
...	P	...																															
...	P	...																															

	Frame Configuration
	Filter Configuration
	Message Buffer Control
	Message RAM Configuration
	Updated from received Frame
	Message Buffer Status
	Parity Bit
	unused

Header 1 (word 0)

Write access via **WRHS1**, read access via **RDHS1**:

- Frame ID: Slot counter filtering configuration
- Cycle Code: Cycle counter filtering configuration
- CHA, CHB: Channel filtering configuration
- CFG: Message buffer configuration: receive / transmit
- PPIT: Payload Preamble Indicator Transmit
- XMI: Transmit mode configuration: single-shot / continuous
- MBI: Message buffer receive / transmit service request enable

Header 2 (word 1)

Write access via **WRHS2**, read access via **RDHS2**:

- Header CRC
 - Transmit Buffer: Configured by the Host (calculated from frame header segment)
 - Receive Buffer: Updated from received frame
- Payload Length Configured
 - Length of data section (2-byte words) as configured by the Host
- Payload Length Received
 - Length of payload segment (2-byte words) stored from received frame

Header 3

Write access via **WRHS3**, read access via **RDHS3**:

- Data Pointer
 - Pointer to the beginning of the corresponding data section in the data partition

Read access via **RDHS3**, valid for receive buffers only, updated from received frames:

- Receive Cycle Count: Cycle count from received frame
- RCI: Received on Channel Indicator
- SFI: Startup Frame Indicator
- SYN: Sync Frame Indicator
- NFI: Null Frame Indicator
- PPI: Payload Preamble Indicator
- RES: Reserved bit

Message Buffer Status MBS (word 3)

Read access via MBS, updated by the Communication Controller at the end of the configured slot.

- VFRA: Valid Frame Received on channel A
- VFRB: Valid Frame Received on channel B
- SEOA: Syntax Error Observed on channel A
- SEOB: Syntax Error Observed on channel B
- CEOA: Content Error Observed on channel A
- CEOB: Content Error Observed on channel B
- SVOA: Slot boundary Violation Observed on channel A
- SVOB: Slot boundary Violation Observed on channel B
- TCIA: Transmission Conflict Indication channel A
- TCIB: Transmission Conflict Indication channel B
- ESA: Empty Slot Channel A
- ESB: Empty Slot Channel B
- MLST: Message LoST
- FTA: Frame Transmitted on Channel A
- FTA: Frame Transmitted on Channel B
- Cycle Count Status: Actual cycle count when status was updated
- RCIS: Received on CHannel Indicator Status
- SFIS: Startup Frame Indicator Status
- SYNS: SYNC Frame Indicator Status
- NFIS: Null Frame Indicator Status
- PPIS: Payload Preamble Indicator Status
- RESS: Reserved Bit Status

2.5.12.2 Data Partition

The data partition of the Message RAM stores the data sections of the message buffers configured for reception / transmission as defined in the header partition. The number of data bytes for each message buffer can vary from 0 to 254. To optimize the data transfer between the shift registers of the two FlexRay Protocol Controllers and the Message RAM as well as between the Host interface and the Message RAM, the physical width of the Message RAM is set to 4 bytes plus one parity bit.

The data partition starts after the last word of the header partition. When configuring the message buffers in the Message RAM the programmer has to assure that the data pointers point to addresses within the data partition. [Table 2-25](#) below shows an example how the data sections of the configured message buffers can be stored in the data partition of the Message RAM.

The beginning and the end of a message buffer's data section is determined by the data pointer and the payload length configured in the message buffer's header section,

FlexRay Protocol Controller (E-Ray)

respectively. This enables a flexible usage of the available RAM space for storage of message buffers with different data length.

If the size of the data section is an odd number of 2-byte words, the remaining 16 bits in the last 32-bit word are unused (see [Table 2-25](#) below)

Table 2-25 Example for Structure of the Data Section in the Message RAM

Bit Word	3	3	3	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
...	P	unused						unused						unused						unused									
...	P	unused						unused						unused						unused									
...	P	MB1 Data3						MB1 Data2						MB1 Data1						MB1 Data0									
...	P									
...	P									
...	P	MB1 Data(n)						MB1 Data(n-1)						MB1 Data(n-2)						MB1 Data(n-3)									
...	P									
...	P									
...	P									
...	P	MB1 Data3						MB1 Data2						MB1 Data1						MB1 Data0									
...	P									
...	P	MB1 Data(k)0						MB1 Data(k-1)0						MB1 Data(k-2)0						MB1 Data(k-3)0									
2046	P	MB80 Data3						MB80 Data2						MB80 Data2						MB80 Data0									
2047	P	unused						unused						MB80 Data5						MB80 Data4									

2.5.12.3 Parity Check

There is a parity checking mechanism implemented in the E-Ray module to assure the integrity of the data stored in the seven RAM blocks of the module. The RAM blocks have a parity generator / checker attached as shown in [Figure 2-25](#). When data is written to a RAM block, the local parity generator generates the parity bit. The E-Ray module uses an even parity (with an even number of one's in the 32-bit data word a zero parity bit is generated). The parity bit is stored together with the respective data word. The parity is checked each time a data word is read from any of the RAM blocks. The module internal data buses have a width of 32 bits.

If a parity error is detected, the respective error flag is set. The parity error flags [MHDS.PIBF](#), [MHDS.POBF](#), [MHDS.PMR](#), [MHDS.PTBF1](#), [MHDS.PTBF2](#), and the faulty message buffer indicators [MHDS.FMBD](#), [MHDS.MFMB](#), [MHDS.FMB](#) are located in the

FlexRay Protocol Controller (E-Ray)

Message Handler Status register. These single error flags control the error interrupt flag **EIR.PERR**.

Figure 2-25 shows the data paths between the RAM blocks and the parity generators / checkers.

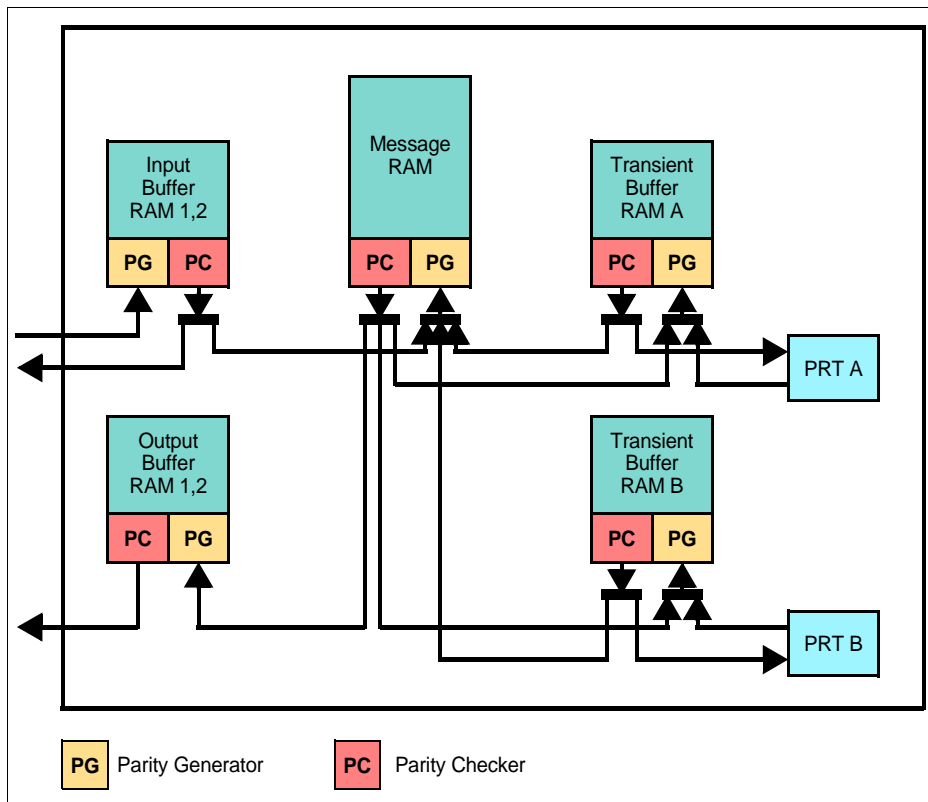


Figure 2-25 Parity Generation and Check

Note: Parity generator & checker are not part of the RAM blocks, but of the RAM access hardware which is part of the E-Ray core.

When a parity error has been detected the following actions will be performed:

In all cases

- The respective parity error flag in the Message Handler Status **MHDS** register is set
- The parity error flag **EIR.PERR** in the Error Service Request Register is set, and if enabled, a module service request to the Host will be generated.

Additionally in specific cases

1. Parity error in data transfer from Input Buffer RAM 1,2 \Rightarrow Message RAM
(Transfer of header and data section)
 - a) **MHDS.PIBF** bit is set
 - b) **MHDS.FMBD** bit is set to indicate that **MHDS.FMB** has been updated
 - c) **MHDS.FMB** indicates the number of the faulty message buffer
 - d) Transmit buffer: Transmission request for the respective message buffer is not set
2. Parity error in data transfer from Input Buffer RAM 1,2 \Rightarrow Message RAM
(Transfer of data section only)
 - a) **MHDS.PMR** bit is set
 - b) **MHDS.FMBD** bit is set to indicate that **MHDS.FMB** points to a faulty message buffer
 - c) **MHDS.FMB** indicates the number of the faulty message buffer
 - d) The data section of the respective message buffer is not updated
 - e) Transmit buffer: Transmission request for the respective message buffer is not set
3. Parity error during host reading Input Buffer RAM
 - a) • **MHDS.PIBF** bit is set
4. Parity error during scan of header sections in Message RAM
 - a) **MHDS.PMR** bit is set
 - b) **MHDS.FMBD** bit is set to indicate that **MHDS.FMB** points to a faulty message buffer
 - c) **MHDS.FMB** indicates the number of the faulty message buffer
 - d) Ignore message buffer (message buffer is skipped)
5. Parity error during data transfer from Message RAM \Rightarrow Transient Buffer RAM A, B
 - a) **MHDS.PMR** bit is set
 - b) **MHDS.FMBD** bit is set to indicate that **MHDS.FMB** points to a faulty message buffer
 - c) **MHDS.FMB** indicates the number of the faulty message buffer
 - d) Frame not transmitted, frames already in transmission are invalidated by setting the frame CRC to zero
6. Parity error during data transfer from Transient Buffer RAM A, B \Rightarrow Protocol Controller 1, 2
 - a) **MHDS.PTBF1**, **MHDS.PTBF2** bit is set
7. Parity error in data transfer from Transient Buffer RAM A, B \Rightarrow Message RAM
(Parity error when reading header section of respective message buffer from Message RAM)
 - a) **MHDS.PMR** bit is set
 - b) **MHDS.FMBD** bit is set to indicate that **MHDS.FMBFMB** points to a faulty message buffer
 - c) **MHDS.FMB** indicates the number of the faulty message buffer
 - d) The data section of the respective message buffer is not updated

FlexRay Protocol Controller (E-Ray)

8. Parity error in data transfer from Transient Buffer RAM A, B \Rightarrow Message RAM
(Parity error when reading Transient Buffer RAM A, B)
 - a) **MHDS.PTBF1**, **MHDS.PTBF2** bit is set
 - b) **MHDS.FMBD** bit is set to indicate that **MHDS.FMBFMB** points to a faulty message buffer
 - c) **MHDS.FMB** indicates the number of the faulty message buffer
9. Parity error during data transfer from Message RAM \Rightarrow Output Buffer RAM
 - a) **MHDS.PMR** bit is set
 - b) **MHDS.FMBD** bit is set to indicate that **MHDS.FMB** points to a faulty message buffer
 - c) **MHDS.FMB** indicates the number of the faulty message buffer
10. Parity error during Host reading Output Buffer RAM
 - a) • **MHDS.POBF** bit is set
11. Parity error during data read of Transient Buffer RAM A, B

When a parity error occurs when the Message Handler reads a frame with network management information (PPI = 1) from the Transient Buffer RAM A, B the corresponding network management vector register NMV1...3 is not updated from that frame.

2.6 Module Service Request

In general, service requests provide a close link to the protocol timing as they are triggered almost immediately when an error or status change is detected by the controller, a frame is received or transmitted, a configured timer service request is activated, or a stop watch event occurred. This enables the Host to react very quickly on specific error conditions, status changes, or timer events. On the other hand too many service requests can cause the Host to miss deadlines required for the application. Therefore the Communication Controller supports disable / enable controls for each individual service request source separately.

An service request may be triggered when

- An error was detected
- A status flag is set
- A timer reaches a preconfigured value
- A message transfer from Input Buffer to Message RAM or from Message RAM to Output Buffer has completed
- A stop watch event occurred

Tracking status and generating service requests when a status change or an error occurs are two independent tasks. Regardless of whether an service request is enabled or not, the corresponding status is tracked and indicated by the Communication Controller. The Host has access to the actual status and error information by reading the Error Service Request Register **EIR** and the Status Service Request **SIR** Register.

Table 2-26 Module Service Request Flags and Service Request Line Enable

Register	Bit	Function
SIR	WST	Wakeup Status
	CAS	Collision Avoidance Symbol
	CYCS	Cycle Start Service Request
	TXI	Transmit Service Request
	RXI	Receive Service Request
	RFNE	Receive FIFO not Empty
	RFF	Receive FIFO Full
	NMVC	Network Management Vector Changed
	TI0	Timer Service Request 0
	TI1	Timer Service Request 1
	TIBC	Transfer Input Buffer Completed
	TOBC	Transfer Output Buffer Completed
	SWE	Stop Watch Event
	SUCS	Startup Completed Successfully
	MBSI	Message Buffer Status Interrupt
	SDS	Start of Dynamic Segment
	WUPA	Wakeup Pattern Channel A
	MTSA	MTS Received on Channel A
	WUPB	Wakeup Pattern Channel B
	MTSB	MTS Received on Channel B
ILE	EINT0	Enable Service Request Line 0
	EINT1	Enable Service Request Line 1

FlexRay Protocol Controller (E-Ray)

Table 2-26 Module Service Request Flags and Service Request Line Enable
(cont'd)

Register	Bit	Function
EIR	PEMC	Protocol Error Mode Changed
	CNA	Command Not Valid
	SFBM	Sync Frames Below Minimum
	SFO	Sync Frame Overflow
	CCF	Clock Correction Failure
	CCL	CHI Command Locked
	PERR	Parity Error
	RFO	Receive FIFO Overrun
EIR	EFA	Empty FIFO Access
	IIBA	Illegal Input Buffer Access
	IOBA	Illegal Output Buffer Access
	MHF	Message Handler Constraints Flag
	EDA	Error Detected on Channel A
	LTVA	Latest Transmit Violation Channel A
	TABA	Transmission Across Boundary Channel A
	EDB	Error Detected on Channel B
	LTVB	Latest Transmit Violation Channel B
	TABB	Transmission Across Boundary Channel B

The interrupt lines to the Host, `eray_int0` and `eray_int1`, are controlled by the enabled interrupts. In addition each of the two interrupt lines can be enabled / disabled separately by programming bit **ILE.EINT0** and **ILE.EINT1**.

The two timer service requests generated by service request timer 0 and 1 are available on pins **eray_tint0** and **eray_tint1**. They can be configured via the Timer 0 and Timer 1 Configuration register.

A stop watch event may be triggered via input pin **eray_stpwt**.

The status of the data transfer between IBF / OBF and the Message RAM is signalled on pins **eray_ibusy** and **eray_obusy**. When a transfer has completed bit **SIR.TIBC** or **SIR.TOBC** is set.

2.7 Restrictions

The following restrictions have to be considered when programming the E-Ray IP-module. A violation of these restrictions may lead to an erroneous behavior of the E-Ray IP-module.

2.7.1 Message Buffers with the same Frame ID

If two or more message buffers are configured with the same frame ID, and if they have a matching cycle counter filter value for the same slot, then the message buffer with the lowest message buffer number is used.

Sharing of a static time slot via cycle counter filtering between different nodes of a FlexRay network is **not** allowed.

2.7.2 Data Transfers between IBF / OBF and Message RAM

The time required to transfer the contents of a message buffer between IBF / OBF and Message RAM depends on the setup time to start the first transfer, the number of 4-byte words to be transferred, and the number of concurrent tasks to be managed by the Message Handler. The number of 4-byte words varies from 4 (header section only) to 68 (header + maximum data section) while the number of concurrent task varies from one to three.

The following concurrent tasks are executed under control of the Message Handler:

- Data transfer between IBF or OBF and Message RAM
- Data transfer between TBF1 and Message RAM, search next TX / RX message buffer CHA
- Data transfer between TBF2 and Message RAM, search next TX / RX message buffer CHB

Transfers between IBF and Message RAM respectively Message RAM and OBF can only be handled one after another. In case that e.g. a transfer between IBF and Message RAM has been started shortly before a transfer between Message RAM and OBF is requested, the OBF transfer has to wait until the IBF transfer has completed.

The relative time is measured in f_{CLC_ERAY} cycles. Absolute time depends on the actual f_{CLC_ERAY} cycle period.

$cyclestrans = (remaining\ cycles\ of\ transfer\ running) + (cycles\ of\ requested\ transfer)$

$cyclestrans = cyclesrem + cyclesreq$

$cyclesrem = (number\ of\ concurrent\ tasks) * (setup\ time + (number\ of\ 4\text{-byte\ words})rem)$

$cyclesreq = (number\ of\ concurrent\ tasks) * (setup\ time + (number\ of\ 4\text{-byte\ words})req)$

$setup\ time = 2 f_{CLC_ERAY} \text{ cycles}$

Under worst case conditions a transfer is requested directly after the previous transfer started:

FlexRay Protocol Controller (E-Ray)

Max. number of $f_{\text{CLC_ERAY}}$ cycles: $\text{cyclestrans} = (3 * (2 + 68)) + (3 * (2 + 68)) = 420$

Worst case timing: $\text{timetrans}(40\text{MHz}) = 420 * 25\text{ns} = 10.5 \text{ ms}$

2.8 Known non functional Features of E-Ray Module Revision 1.0.0

The implemented E-Ray IP Block does not contain all function described throughout this document.

Missing cycle start interrupt in startup phase.

When communication is restarted by CHI command RUN after the Communication Controller left "STARTUP", "NORMAL_ACTIVE", or "NORMAL_PASSIVE" state by application of CHI command READY it may happen, that the cycle start interrupt flag **SIR.CYCS** is not set in

- cycle 0 in case of a leading coldstarter
- cycles 0-2 in case of an integrating node

of the following startup phase (see FlexRay Protocol Spec v2.1, fig. 7-10).

The erratum is limited to applications where READY command is used to leave "STARTUP", "NORMAL_ACTIVE", or "NORMAL_PASSIVE" state.

Cycle start interrupt is generated correctly for all following cycles.

Leading coldstarter: In cycle 0 of the startup phase no cycle start interrupt is generated

Integrating node: In cycles 0-2 of the startup phase no cycle start interrupt is generated.

Workaround: Don't leave "STARTUP", "NORMAL_ACTIVE", or "NORMAL_PASSIVE" state by application of CHI command READY

Update of status registers **CCEV** and **CCSV** delayed with respect to interrupt flags **EIR.PEMC**, **SIR.WST**, **SIR.SUCS**, **SIR.WUPA**, **SIR.WUPB**.

A change of Error Mode, Protocol Operation Control Status, and Wakeup Status sets the interrupt flags **EIR.PEMC**, **SIR.WST**, **SIR.SUCS**, **SIR.WUPA**, and **SIR.WUPB**. The status registers are copied into the Host clock domain (**CCEV**, **CCSV**) to allow read access. This clock domain crossing delays the update of the respective status fields **CCEV.ERRM**, **CCSV.POCS**, and **CCSV.WSV** by up to 5 cycles of the slower of the two clocks **eray_bclk** and **eray_sclk** in relation to the change of the interrupt flags.

The erratum is limited to applications where evaluation of the status fields **CCEV.ERRM**, **CCSV.POCS**, and **CCSV.WSV** is triggered by a change of one of the interrupt flags **EIR.PEMC**, **SIR.WSV**, **SIR.SUCS**, **SIR.WUPA**, or **SIR.WUPB**.

In case the Host reads the Error Mode **CCEV.ERRM**, the Protocol Operation Control Status **CCSV.POCS**, or the Wakeup Status **CCSV.WSV** directly after the respective interrupt flag was set, it may happen that registers **CCEV** respectively **CCSV** are not yet updated.

FlexRay Protocol Controller (E-Ray)

Workaround: Delay reading of status fields **CCEV.ERRM**, **CCSV.POCS**, or **CCSV.WSV** for at least 6 cycles of the slower of the two clocks `eray_bclk` and `eray_sclk` after the respective interrupt flag was set.

Clearing of interrupt flags **EIR.PEMC, **SIR.WST**, **SIR.WUPA**, **SIR.WUPB**, **SIR.CAS**, and **SIR.SUCS** not accepted.**

A change of POC Error Mode, Wakeup Status, reception of a Collision Avoidance Symbol, or a successfully completed startup sets some of the interrupt flags **EIR.PEMC**, **SIR.WST**, **SIR.WUPA**, **SIR.WUPB**, **SIR.CAS**, and **SIR.SUCS**. Because the set condition for the respective interrupt flag may be active for up to 4 `eray_bclk` + 4 `eray_sclk` cycles, it can happen that the flag is not reset to 0 if the Host clears the flag directly after it has been set.

The erratum is limited to applications where the interrupt flags **EIR.PEMC**, **SIR.WST**, **SIR.WUPA**, **SIR.WUPB**, **SIR.CAS**, and **SIR.SUCS** are used for interrupt generation.

In case the Host clears one of the affected flags, it may happen that the interrupt stays active. A following read access will return 1 for the respective flag.

Workaround: After clearing of one of the affected interrupt flags, the Host has to re-read the flag. If the flag is still set, the Host has to repeat the sequence until it reads a 0.

Content error detected in sync frame at slot boundary.

In case a slot is configured for sync frame reception and a valid sync frame was received on one channel while a content error is detected exactly at the last `sclk` of that slot on the other channel, the valid sync frame is used for clock synchronization. For this case the FlexRay protocol specification requires that the valid frame is not used for clock synchronization. See FlexRay protocol specification v2.1, Fig. 6-8 on page 135. If the content error on the other channel is detected at least one `sclk` earlier, the valid sync frame is not used for clock synchronization (correct behavior). If the frame on the other channel is decoded one `sclk` later, this results in a boundary violation, not a content error; the valid sync frame is used for clock synchronization (also correct behavior).

The erratum is limited to cases where a valid sync frame is received on one channel while a frame with content error is decoded at the last `sclk` of the slot on the other channel.

In the described case the valid sync frame is used for clock synchronization.

Workaround: Low, probability for content error detection at last `sclk` of slot is low. The used sync frame is probably valid, otherwise it is filtered by the fault tolerant midpoint algorithm (see FlexRay protocol specification v2.1, section 8.6.1). Workaround not necessary.

Loop back mode operates only at 10 MBit/s.

The looped back data is falsified at the two lower baud rates of 5 and 2.5 MBit/s.

FlexRay Protocol Controller (E-Ray)

The erratum is limited to test cases where loop back is used with the baud rate prescaler (**PRTC1.BRP**) configured to 5 or 2.5 MBit/s.

The loop back self test is only possible at the highest baud rate.

Workaround: Run loop back tests with 10 MBit/s (**PRTC1.BRP** = 00_B).

False clock correction value possible in case action point coincides with end of received sync frame.

In case the action point coincides with the end of a received sync frame, the relative deviation value (= time between secondary time reference point and action point) of this particular sync frame is seen as zero.

Critical Case: When a valid sync frame is received in a specific time relation to the node's internal action point (internal signal prt_frame_decoded is high one sclk period after high pulse of gtu_action_point). Not critical when the frame is decoded one sclk before or one sclk after this critical case.

The erratum is limited to cases where the frame length is configured to a smaller value than the action point offset, and the complete frame is received between start of the slot and the slot's action point.

In the described case a relative deviation value of zero is calculated from the received valid sync frame. This may result in a lower absolute value for the next offset and/or rate correction.

Workaround: Avoid configurations with frame length smaller than action point offset.

Noise following a dynamic frame that delays idle detection may fail to stop slot counting for the remainder of the dynamic segment.

If (in case of noise) the time between "potential idle start on X" and "CHIRP on X" (see Protocol Spec. v2.1, Figure 5-21) is greater than gdDynamicSlotIdlePhase, the E-Ray will not remain for the remainder of the current dynamic segment in the state 'wait for the end of dynamic slot rx'. Instead, the E-Ray continues slot counting. This may enable the node to further transmissions in the current dynamic segment.

The erratum is limited to noise that is seen only locally and that is detected in the time window between the end of a dynamic frame's DTS and idle detection ("CHIRP on X").

In the described case the faulty node may not stop slot counting and may continue to transmit dynamic frames. This may lead to a frame collision in the current dynamic segment.

Workaround: Problem only occurs at local bus errors in a specific short time window. Even without this erratum no communication is possible for the remainder of the dynamic segment if noise is seen by all nodes. Therefore no workaround.

FlexRay Protocol Controller (E-Ray)**Mismatch between macrotick value and cycle counter value at cycle boundary.**

The macrotick value MTCCV.MTV is updated one eray_blk period before the cycle counter value MTCCV.CCV is updated.

If a stop watch event occurs in the last macrotick of a cycle (in the last 4 eray_blk periods before cycle counter increment), the captured macrotick value STPW1.SMTV and the captured cycle counter value STPW1.SCCV will have the same mismatch.

If the Timer 0 Macrotock Offset T0C.T0MO is configured to zero and the current cycle count is matching the configured Timer 0 Cycle Code T0C.T0CC while timer 0 is in mode "running", a timer 0 interrupt is generated erroneously at the end of the matching cycle.

The erratum is limited to cases where register MTCCV is read one eray_blk period before the cycle counter is incremented.

For the stop watch feature the erratum is limited to the case where the stop watch event occurs in the last macrotick of a cycle (latest 4 eray_blk periods before cycle counter increment).

With respect to timer 0 the erratum is limited to a Timer 0 Macrotock Offset of zero.

In the described cases the values of macrotick and cycle counter in register MTCCV and the captured values of macrotick and cycle counter in register STPW1 may not be consistent. In addition a timer 0 interrupt may be generated erroneously at the end of the cycle matching the configured Timer 0 Cycle Code.

Workaround: Re-read register MTCCV if it was read at the end of a cycle. Consider the reduced time resolution if the stop watch was triggered at the end of a cycle. Don't configure a Timer 0 Macrotock Offset of zero.

Wrong payload data transmitted due to disturbed channel-idle sequence.

If the E-Ray does not detect channel-idle (11 consecutive bit HIGH) between the end of a transmission and the beginning of the next transmission in the following transmit slot due to disturbances on the FlexRay bus, a frame with wrong payload data may be transmitted in the following transmit slot.

Note: Disturbances on the FlexRay bus in the extent needed to reproduce the described behavior are beyond the scope of the FlexRay Conformance Test.

The erratum is limited to the case where no channel-idle sequence is detected between two frames transmitted by the same node in two consecutive static transmit slots.

The frame transmitted in a slot following a transmit slot with completely destroyed channel-idle sequence holds wrong payload data.

Workaround: Problem only occurs due to severe disturbances on the FlexRay bus. In case of such disturbances it is very likely that the transmitted frames are also disturbed and will not be accepted by the receivers.

The probability for occurrence of that problem decreases when the gap between slot boundaries and start of frame resp. end of frame is increased.

FlexRay Protocol Controller (E-Ray)

Status of Network Management Vector NMVn after the Communication Controller has left "NORMAL_ACTIVE" or "NORMAL_PASSIVE" state by application of commands READY, HALT, or FREEZE.

When communication is restarted after the Communication Controller has left "NORMAL_ACTIVE" or "NORMAL_PASSIVE" state by application of commands READY, HALT, or FREEZE, it may happen that NMVn holds invalid data before re-entering "NORMAL_ACTIVE" state.

The erratum is limited to cases where NMVn is read by the Host when the Communication Controller is outside "NORMAL_ACTIVE" or "NORMAL_PASSIVE" state.

In the described case the Host may read data from NMVn which originates from previous communication cycles.

Workaround: Don't evaluate the Network Management Vector NMVn outside "NORMAL_ACTIVE" or "NORMAL_PASSIVE" state. Problem only appears outside NORMAL_ACTIVE or NORMAL_PASSIVE state.

Reception of wakeup pattern signalled for channel disconnected by **SUCC1.CCHA/B = 0**.

In case the Communication Controller is physically connected to a two channel network while one of the two channels is disabled by the Host by programming **SUCC1.CCHA/B = 0**, the Communication Controller will signal the reception of a wakeup pattern on that channel by **SIR.WST = 1** and **CCSV.WSV = 010_B** (RECEIVED_WUP).

The erratum is limited to cases where the Communication Controller is physically connected to both channels of a two channel network and one of the two channels is configured to be disconnected by the Host by programming **SUCC1.CCHA/B = 0**.

The relative deviation time between secondary time reference point and action point of the offset and/or rate correction value is calculated. This wrong deviation value may be used by the FTM algorithm to calculate the midpoint. Only in this case a wrong offset and/or rate correction value is calculated.

Workaround: No Workaround, but the affected configuration is not a realistic application. No problem in single channel applications.

Clock correction value calculated with wrong deviation value.

In case of receiving a valid frame after detecting an invalid frame before the action point, the clock correction value is calculated with a wrong deviation value.

The erratum is limited to the case where an invalid frame (at least valid TSS, FSS, and BSS) is detected before the action point and a valid frame, starting after the action point, is received in the same slot.

FlexRay Protocol Controller (E-Ray)

The relative deviation (= time between secondary time reference point and action point) of the invalid frame is stored for the following rate and offset correction value calculation. This wrong deviation value may be used by the FTM algorithm to calculate the midpoint. Only in this case a wrong offset and/or rate correction value is calculated.

Workaround: No Workaround, but the probability of noise leading to detect a frame start or an invalid frame before the action point followed by a valid frame is very low. Due to the FTM algorithm the impact of a single faulty deviation value is low.

Wrong payload data written to receive buffer.

When a receive slot is directly followed by a transmit slot and when the eray_bclk frequency is below the minimum frequency required for the configured action point offset as listed in the table below, it may happen, that a word of the received payload is stored twice into the respective receive buffer in the Message RAM while all following words are shifted by one address, and the last word is lost.

With the maximum payload of 254 byte and both channels used (SUCC1.CCHA/B = 1) the problem may appear, depending on the configured (M)APO and TSST, if the eray_bclk is below the minimum value listed in the table below.

In case only one channel is used (SUCC1.CCHA or B = 0) the values above can be multiplied by 0.66. With lower payload values the minimum eray_bclk also decreases.

Table 2-27 Wrong Payload Data Configuration

APO: Action Point Offset MAPO: Minislot Action Point Offset	minimum eray_bclk, TSST = 10	minimum eray_bclk, TSST = 3
1	63 MHz	70 MHz
2	48 MHz	53 MHz
3	40 MHz	43 MHz
4	34 MHz	36 MHz
5	30 MHz	31 MHz

In case only one channel is used (SUCC1.CCHA or B = 0) the values above can be multiplied by 0.66. With lower payload values the minimum eray_bclk also decreases.

The erratum is limited to the case where eray_bclk is below a minimum frequency. This frequency depends on the configuration of GTUC9.APO, GTUC9.MAPO, and PRTC1.TSST as well as on the configured payload.

The payload of the affected receive buffer is falsified.

Workaround: Configure (M)APO and TSST according to table above depending on the used eray_bclk frequency.

FlexRay Protocol Controller (E-Ray)

Problem does not appear with (M)APO configurations used in actual applications and the `eray_bclk` frequencies implemented in the available designs.

In case of a faulty configuration of pLatestTx and transmission across dynamic segment boundary error flags EIR.LTVA/B may not be set when a latest transmit violation occurs.

Prerequisites:

- Faulty configuration of pLatestTx (MHDC.SLT) does not prevent transmission of frame X in cycle n and frame Y in cycle n+1.
- The last dynamic frame X transmitted in cycle n ends in minislot m.

Fault Case: In case frame Y is the only frame transmitted in the dynamic segment of cycle n+1 and frame Y is transmitted across the end of dynamic segment and the last minislot of dynamic segment has the value m, error flags EIR.LTVA/B (Latest Transmit Violation Channel A,B) are not set while EIR.TABA/B (Transmission Across Boundary Channel A,B) are set as specified.

The erratum is limited to the case where a transmission across dynamic segment boundary is not prevented because of a faulty configuration of pLatestTx (MHDC.SLT).

In case of transmission across dynamic segment boundary the CC enters POC state HALT and the error flags EIR.TABA or EIR.TABB are set. In case of a faulty configuration of pLatestTx, the flag EIR.LTVA or EIR.LTVB are not set.

Workaround: Configure pLatestTx as required by the FlexRay protocol specification v2.1. Problem appears only with faulty configuration of pLatestTx.

In case `eray_bclk` is below `eray_sclk/2`, TEST1.CERA/B may fail to report a detected coding error.

All detected coding errors should be reported in the Test Register 1, at TEST1.CERA/B. If `eray_bclk` is below `eray_sclk/2`, it may happen, depending on phase difference of `eray_bclk` and `eray_sclk`, that TEST1.CERA/B are not updated in case of a detected coding error and remain in the state `0000B` = "No coding error detected".

The erratum is limited to the case where `eray_bclk` is below `eray_sclk/2`.

Coding errors not reported via TEST1.CERA/B. The frame decoding is not affected.

Workaround: No Workaround, but only hardware test function affected.

Generating EIR.SFO considers number of sync frames in the last cycle only, double cycle not evaluated.

In case that there are different sync IDs received for the two cycles of a double cycle, and the total number of sync frames with different IDs received in the double cycle exceeds the maximum number of sync frames as configured by GTUC2.SNM[3:0] while

FlexRay Protocol Controller (E-Ray)

the number of sync frames within each cycle of the double cycle is below GTUC2.SNM, the sync frame overflow indication EIR.SFO is not set.

The erratum is limited to the case where sync frames with different IDs are received in even and odd cycles and where the total number of sync frames is greater than GTUC2.SNM.

No sync frame overflow signalled if number of received sync frames in each of the two cycles of a double cycle is below the maximum number of sync frames as configured by GTUC2.SNM.

Workaround: No Workaround, but only appears in case of GTUC2.SNM lower than the number of sync nodes configured in a cluster.

CAS collision in case of macrotick length > CAS.

A leading coldstarter that has switched from state "COLDSTART_LISTEN" to "COLDSTART_COLLISION_RESOLUTION" and that receives a CAS symbol transmitted by another coldstarter in the time window of cCASAActionPointOffset (1 MT) after the state change will transmit a CAS symbol at the CAS action point. This CAS symbol should have been suppressed.

The erratum is limited to the case where the macrotick is configured to be longer than the CAS symbol.

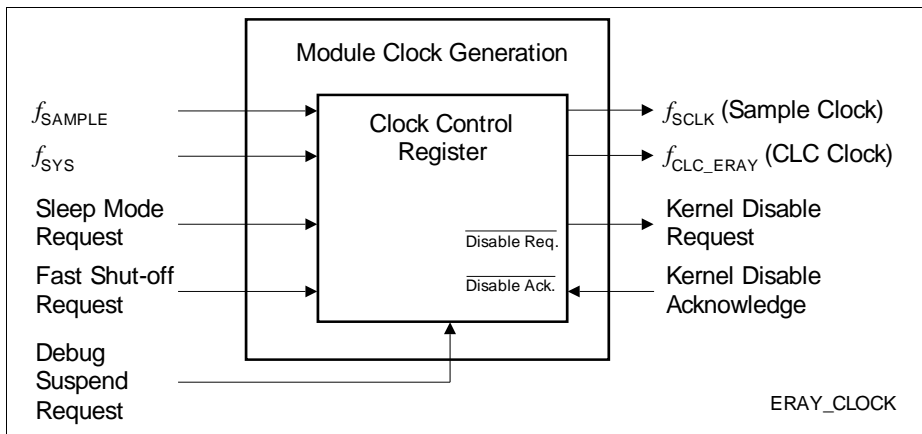
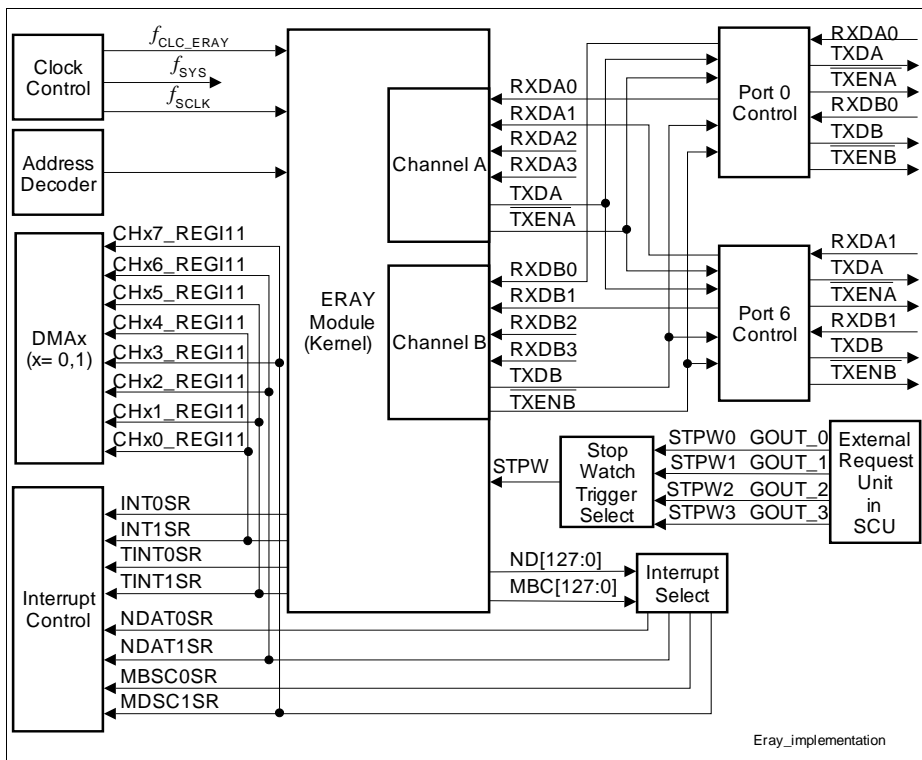
A CAS collision will disturb the first cycle of the startup, delaying the startup success by one cycle.

Workaround: No Workaround, but in actual applications the macrotick length is configured to be shorter than CAS. Low probability for two cold starters to transmit a CAS almost simultaneously.

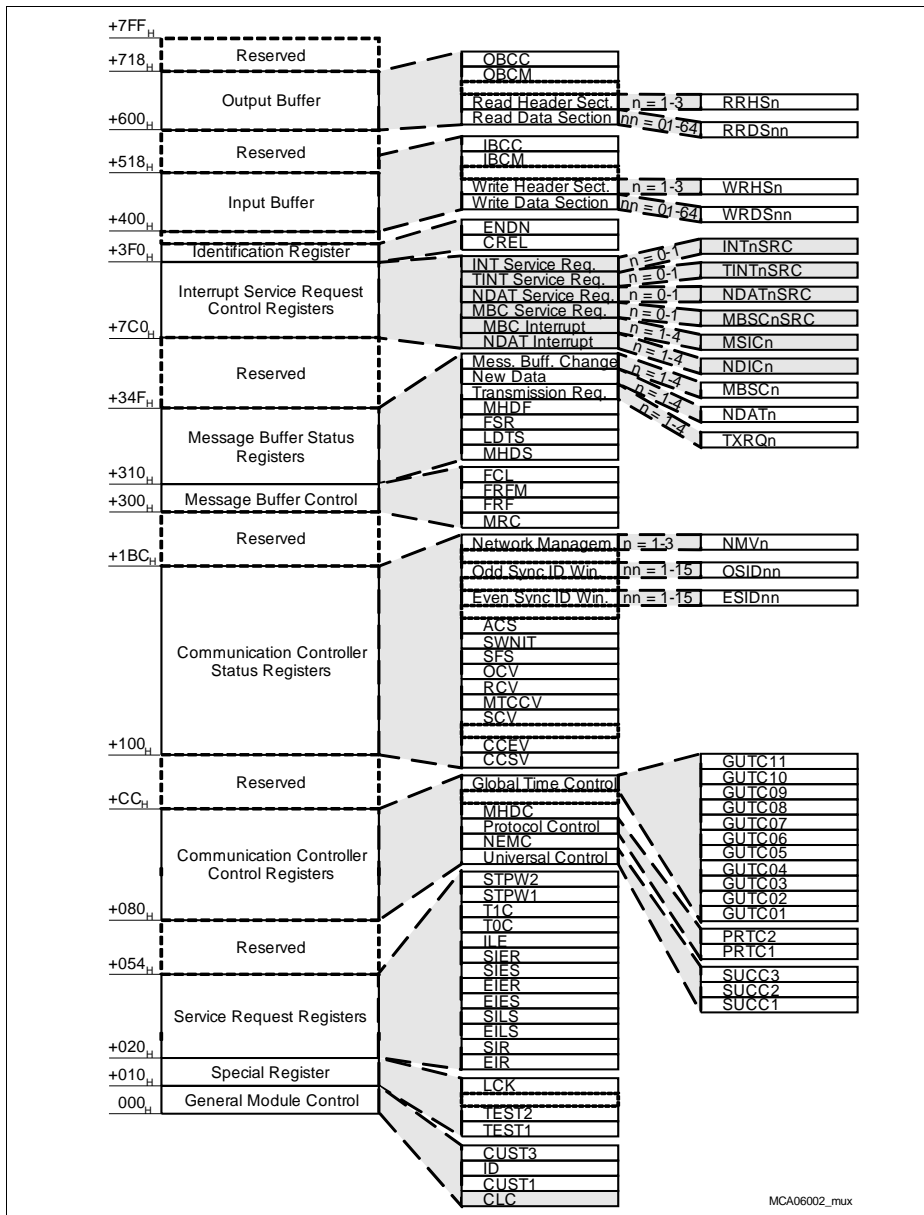
SYS_SAMPLE STCU_SYSCON.ENERAY

To trigger the DMA, the eray_obusy signal is routed to the DMA. An active eray_obusy signal indicates an ongoing transfer from E-Ray Message Buffer to E-Ray Output Buffer.

FlexRay Protocol Controller (E-Ray)



FlexRay Protocol Controller (E-Ray)



MCA06002_mux

2.9 Revision History

The current revision of Bosch Specification this User's Manual is based on is: **Revision 1.2.3u.**

Table 2-28 Revision History

Version Number	Changes to Previous Version
Rev_0.1	First Draft
Rev_0.2	Adapted to PWD 0.8
Rev_0.3	Chapters 3, 5, 6 completed
Rev_0.4	Adapted to actual state of protocol development
Rev_0.5	Adapted to actual state of protocol development
Rev_0.51	Adapted to actual state of protocol development
Rev_0.52	Adapted to actual state of protocol development
Rev_0.53	Adapted to actual state of protocol development
Rev_0.6	Adapted to actual state of protocol development
Rev_0.61	Adapted to actual state of protocol development
Rev_0.62	Adapted to actual state of protocol development
Rev_1.0	First complete revision
Rev_1.01	Message Buffer Status bits PLE, MLST, ES replaced by bits ESA, ESB, MLST
Rev_1.02	IBCR, IBCM, OBCR, OBCM: addresses changed HDC2: register removed MHDC1: renamed to MHDC Message buffer 0 dedicated to hold key slot ID SFS: description updated ESIDn, OSIDn: description updated EIR: bit SCE removed EILS: bit SCEL removed EIES, EIER: bit SCEE removed

FlexRay Protocol Controller (E-Ray)

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_1.2	<p>IDEFAULT_CONFIG added to POC working</p> <p>CCSV: assignment of states to ^{POCS} changed:</p> <p>POCS = 00 0000_B = "DEFAULT_CONFIG"</p> <p>POCS = 00 1111_B = "CONFIG"</p> <p>CCSV: bit DCREQ removed</p> <p>SIR: bit MBSI added</p> <p>SILS: bit MBSIL added</p> <p>SIES, SIER: bit MBSIE added</p> <p>Register BGSC removed</p> <p>EIR: bits SMEB, SMEA removed</p> <p>EILS: bits SMEBL, SMEAL removed</p> <p>EIES, EIER: bits SMEBE, SMEAE removed</p> <p>Registers TXRQ3, TXRQ4, NDAT3, NDAT4, MBSC3, MBSC4 added</p> <p>Bus guardian related pins eray_arm, eray_bgt, eray_mt, eray_bge1, and eray_bge2 have no function</p> <p>PRTC1: Configuration parameter CASM added</p> <p>WRHS1: Bit NME changed to PPIT</p> <p>RDHS1: Bit NME changed to PPIT</p> <p>Pin eray_scanmode for scan mode control added</p>
Rev_1.3	<p>Changed "r" bits into "rh" bits.</p> <p>Made ACS Register writable.</p> <p>Included Reserved Bits into EIES and EIER</p> <p>Changed access write from "rw" to "rwh" for Register CUST1.IEN. Changed access type of unused bits (0) to "rw".</p> <p>Changed access type of CUST1.INT0 to rwh.</p> <p>Changed Reset value of CUST0 to C104 0105_H</p> <p>Updated the "Known Limitations of E-Ray IP-Module", revision 4.07.2005.</p>

FlexRay Protocol Controller (E-Ray)

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_1.4	<p>Updated the "Errata of E-Ray IP-Module", revision 1.0 30.06.2006.</p> <p>Renamed from "Interrupt Flag Interface" to "Internal Signal and Flag Interface"</p> <p>With this revision it is possible to use message buffer 1 for sync frame transmission in addition to message buffer 0 if sync frames should have different payloads on channel A and B</p> <p>TEST1: Bit ELBE for control of internal / external loop back mode added, description of internal loop back added</p> <p>EIR: Handling of bits PERR and RFO same as for other bits, bit MHF added</p> <p>SIR: Bit RFF renamed to RFCL, handling of bits RFNE, RFCL same as for other bits</p> <p>EILS: Bit MHFL added,</p> <p>SILS: Bit RFFL renamed to RFCLL</p> <p>EIES, EIER: Bit MHFE added</p> <p>SIES, SIER: Bit RFFE renamed to RFCLE</p> <p>Register STPW renamed to STPW1</p> <p>STPW2: Register added</p> <p>CCSV: Bits PSL added</p> <p>SWNIT: Bits MTSA, MTSB added</p> <p>MRC: Bit SPLM added</p> <p>FCL: Register added</p> <p>FSR: Register added</p> <p>MHDF: Register added</p> <p>MBSC1/2/3/4: Naming of bits changed from MBS to MBC to distinguish between message buffer status flag (MBC) and message buffer status register (MBS)</p> <p>CREL: Register added</p> <p>ENDN: Register added</p> <p>Message buffers in Message RAM: Header 2 and 3 updated from received data frames only</p> <p>MBS: Bits FTA, FTB, CCS, RCIS, SFIS, SYNS, NFIS, PPIS, RESS added</p> <p>Description Asynchronous Transmit Mode, added: ...This write operation has to be directly preceded by two consecutive write accesses to the Configuration LockKey (unlock sequence)...</p> <p>Description Loop Back Mode, added:...This write operation has to be directly preceded by two consecutive write accesses to the Configuration LockKey (unlock sequence).</p>

FlexRay Protocol Controller (E-Ray)

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_1.4 (cont'd)	<p>Description LCK.CLK, corrected: To leave CONFIG state by writing SUCC1.CMD (commands READY, MONITOR_MODE, ATM_LOOP_BACK), ...Third write: SUCC1.CMD</p> <p>Description EIR.MHF, added: The flag signals a Message Handler constraints violation condition. It is set whenever one of the flags MHDF.SNUA, MHDF.SNUB, MHDF.FNFA, MHDF.FNFB, MHDF.TBFA, MHDF.TBFB, MHDF.WAHP changes from 0 to 1.</p> <p>Description SIR.CAS, added: This flag is set by the Communication Controller during STARTUP state when a CAS or a potential CAS was received. 1 = Bit pattern matching the CAS symbol received 0 = No bit pattern matching the CAS symbol received.</p> <p>Description SIR.MBSI, corrected: This flag is set by the Communication Controller when the message buffer status MBS has changed if bit MBI of that message buffer is set.</p> <p>Description T0C, added: Note: The configuration of timer 0 is compared against the macrotick counter value, there is no separate counter for timer 0...</p> <p>Description SUCC1.CMD, modified: Note included into description of CHI command CLEAR_RAMs. ...Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAMs...</p> <p>Description SUCC1.TSM, changed: ...The key slot ID is configured in the header section of message buffer 0 respectively message buffers 0 and 1 depending on bit MRC.SPLM. In case TSM = 1, message buffer 0 respectively message buffers 0,1 can be (re)configured in "DEFAULT_CONFIG" or "CONFIG" state only. In ALL slot mode the Communication Controller may transmit in all slots. TSM is a configuration bit which can only be set / reset by the Host. The bit can be written in "DEFAULT_CONFIG" or "CONFIG" state only. The Communication Controller changes to ALL slot mode when the Host successfully applied the ALL_SLOTS command by writing CMD = "0101" in POC states "NORMAL_ACTIVE" or "NORMAL_PASSIVE"...</p> <p>Description PRTC1.SPP, added: Note: The current revision 2.1 of the FlexRay protocol requires that SPP = "00". The alternate strobe point positions could be used to compensate for asymmetries in the physical layer.</p> <p>Description CCSV.CSNI, corrected: ...Reset by CHI command RESET_STATUS_INDICATORS or by transition from "HALT" to "DEFAULT_CONFIG" state or from "READY" to "STARTUP" state.</p>

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_1.4 (cont'd)	<p>Description CCSV.CSAI, corrected: ...Reset by CHI command RESET_STATUS_INDICATORS or by transition from "HALT" to "DEFAULT_CONFIG" state or from "READY" to "STARTUP" state.</p> <p>Description CCSV.WSV, corrected: ...Reset by CHI command RESET_STATUS_INDICATORS or by transition from "HALT" to "DEFAULT_CONFIG" state or from "READY" to "STARTUP" state.</p> <p>Description SFS.VSAE, modified: Holds the number of valid sync frames received on channel A in the even communication cycle. If transmission of sync frames is enabled by SUCC1.TXSY the value is incremented by one...</p> <p>Description SFS.VSAO, modified: Holds the number of valid sync frames received on channel A in the odd communication cycle. If transmission of sync frames is enabled by SUCC1.TXSY the value is incremented by one...</p> <p>Description SFS.VSBE, modified: Holds the number of valid sync frames received on channel B in the even communication cycle. If transmission of sync frames is enabled by SUCC1.TXSY the value is incremented by one...</p> <p>Description SFS.VSBO, modified: Holds the number of valid sync frames received on channel B in the odd communication cycle. If transmission of sync frames is enabled by SUCC1.TXSY the value is incremented by one...</p> <p>Description ACS, added: Note: The set condition of flags CIA and CIB is also fulfilled if there is only one single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase...</p> <p>Description MRC.SEC1:0, changed: ...Exception: In nodes configured for sync frame transmission or for single slot mode operation message buffer 0 (and if SPLM = 1, also message buffer 1) is always locked</p> <p>Description MRC.SPLM, changed: This bit is only evaluated if the node is configured as sync node (SUCC1.TXSY = 1) or for single slot mode operation (SUCC1.TSM = 1)...</p> <p>Description MRC.SPLM, changed: Note: In case the node is configured as sync node (SUCC1.TXSY = 1) or for single slot mode operation (SUCC1.TSM = 1), message buffer 0 resp. 1 is reserved for sync frames or single slot frames and have to be configured with the node-specific key slot ID. In case the node is neither configured as sync node nor for single slot operation message buffer 0 resp. 1 is treated like all other message buffers.</p> <p>Description Parity Check, changed: ...When a parity error occurs when the Message Handler reads a frame with network management information (PPI = 1) from the Transient Buffer RAM A, B the corresponding network management vector register NMV1...3 is not updated from that frame.</p> <p>Included TC1797 O Ports the Physical Layer Interface is connected to.</p>

FlexRay Protocol Controller (E-Ray)

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_1.4 (cont'd)	<p>Description MBS.MLST, added: The flag is set in case the Host did not read the message before the message buffer was updated from a received data frame... The flag is reset by a Host write to the message buffer via IBF or when a new message is stored into the message buffer after the message buffers ND flag was reset by reading out the message buffer via OBF.</p> <p>Description MBS, corrected: Note: The FlexRay protocol specification requires that FTA, and FTB can only be reset by the Host...</p> <p>Description Null Frame Transmission, changed: ...In this case, no message buffer status MBS is updated.</p>
Rev_1.5	<p>Included BPI Register (Service Request Nodes: INT0SRC, INT1SRC, TINT0SRC, TINT1SRC, NDAT0SRC, NDAT0SRC, MBSC0SRC, MBSC1SRC, CLC).</p> <p>Included for New Data and Message Buffer Status Changed Flags eight Interrupt Select Register (NDIC0, NDIC1, NDIC2, NDIC3, MSIC0, MSIC1, MSIC2, MSIC3)</p> <p>Included new address range for TC1797.</p> <p>Included into CUST1 Register Control (IBF1PAG, IBF1PAG) and Status Bits (IBS) for multiple buffer control.</p> <p>Included for TC1797 the IO Ports the Physical Layer Interface is connected to.</p> <p>Included into CUST1 1 out of 4 multiplexer Select Input Control for RXA (CUST1.RISA), RXB (CUST1.RISB) and STPWT (CUST1.STPWTS).</p> <p>Included DMA Trigger connections for TC1797.</p>
Rev_1.6	<p>Included a description of the Delayed Write Scheme to OBF and IBF.</p> <p>Corrected an erroneous description of IBF1PAG and IBF2PAG concerning the write access for specific values of IBFS. Included a verification scheme to verify correct polarity of IBFS and logic for IBF1PAG and IBF2PAG.</p>
Rev_1.7	<p>Changed P6.7 to P6.10 to use only p6.PDR1 only.</p> <p>Included table summarizing access wait states.</p>
Rev_1.8	<p>Included a description of registering the reset when E-Ray module is disabled and executing the reset when it is enabled hereafter.</p>

FlexRay Protocol Controller (E-Ray)

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_1.9	<p>Included IP Specification changes of Revision 1.2.3.</p> <p>Description TEST1, added: When the E-Ray IP is operated in one of its test modes that requires WRTE to be set (RAM Test Mode, I/O Test Mode, Asynchronous Transmit Mode, and Loop Back Mode) only the selected test mode functionality is available. The test functions are not available in addition to the normal operational mode functions, they change the functions of parts of the E-Ray module. Therefore normal operation as specified outside this chapter and as required by the FlexRay protocol specification and the FlexRay conformance test is not possible. Test mode functions may not be combined with each other or with FlexRay protocol functions. The test mode features are intended for hardware testing or for FlexRay bus analyzer tools. They are not intended to be used in FlexRay applications.</p> <p>Description TEST1.ELBE, modified: ...Bit ELBE is evaluated only when POC is in loop back mode and test multiplexer control is in non-multiplexing mode TMC = 00.</p> <p>Description TEST1.TMC, modified: TMC Test Multiplexer Control 00, 11= Normal signal path (default); 01 = RAM Test Mode - Internal busses are multiplexed to make all RAM blocks of the E-Ray module directly accessible by the Host...</p> <p>Description Loop Back Mode, corrected: ...Reading CCSV.POCS will return "00 1101" while the E-Ray module is in loop back mode...</p> <p>Description Loop Back mode, completed: ...When the CC is in loop back mode, a loop back test is started by the Host writing a message to the Input Buffer and requesting the transmission by writing to register IBCR...</p> <p>Description LCK.CLK, corrected: ...If the write sequence below is interrupted by other write accesses between the second write to the Configuration Lock Key and the write access to the SUCC1 register, the Communication Controller remains in CONFIG state and the sequence has to be repeated.</p> <p>Description LCK.TMK, corrected: ...If the write sequence below is interrupted by other write accesses between the second write to the Test Mode Key and the write access to the TEST1 register, TEST1.WRTE is not set to '1' and the sequence has to be repeated.</p> <p>Description EIR.CCL, corrected: The flag signals that the write access to the CHI command vector SUCC1.CMD was not successful because the execution of the previous CHI command has not yet completed...</p> <p>Description SIR.MTSA,B, modified: Media Access Test symbol received on channel A,B during the preceding symbol window...</p>

FlexRay Protocol Controller (E-Ray)

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_1.9 (cont'd)	<p>Description T1C, removed:...In case the configured macrotick count is not within the valid range, timer 1 will not start...</p> <p>Description SUCC1.CMD, corrected: ...In case the previous CHI command has not yet completed, EIR.CCL is set to 1 together with EIR.CNA; the CHI command needs to be repeated. Except for HALT state, a POC state change command applied while the CC is already in the requested POC state will be ignored.</p> <p>Description SUCC1.CMD, added: ALLOW_COLDSTART; ...When called in states DEFAULT_CONFIG, CONFIG, HALT, or MONITOR_MODE, CMD will be reset to 0000 = command_not_accepted...</p> <p>Description SUCC1.CMD, completed: RESET_STATUS_INDICATORS; ...Flags internally evaluated in the actual POC state are not reset...</p> <p>Description SUCC1.CMD, corrected: MONITOR_MODE; ...In this mode the CC is able to receive FlexRay frames and wakeup pattern...</p> <p>Description SUCC1.CMD, added: Table added which references the CHI commands from the FlexRay Protocol Specification v2.1 (section 2.2.1.1, Table 2-2) to the E-Ray CHI command vector CMD.</p> <p>Description CCSV.RCA, completed: ...The READY command resets this counter to the maximum number of coldstart attempts as configured by SUCC1.CSA.</p> <p>Description SWNIT.MTSA,B, modified: Media Access Test symbol received on channel A,B during the preceding symbol window...</p> <p>Description ESID[1...15].RXEA,B completed: ...sorted in ascending order,If the node transmits a sync frame in an even communication cycle by itself, register ESID1 holds the respective sync frame ID as configured in message buffer 0 and flags RXEA, RXEB are set... RXEA,B Received / Configured Even Sync ID on Channel A,B Signals that a sync frame corresponding to the stored even sync ID was received on channel A or that the node is configured to be a sync node with key slot = EID (ESID1 only). 1 = Sync frame received on channel A / node configured to transmit sync frames; 0 = No sync frame received on channel A / node not configured to transmit sync frames.</p>

FlexRay Protocol Controller (E-Ray)

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_1.9 (cont'd)	<p>Description OSID[1...15].RXOA,B completed:...sorted in ascending order, ... If the node transmits a sync frame in an odd communication cycle by itself, register OSID1 holds the respective sync frame ID as configured in message buffer 0 and flags RXOA, RXOB are set...RXOA,B Received / Configured Odd Sync ID on Channel A,B Signals that a sync frame corresponding to the stored odd sync ID was received on channel A or that the node is configured to be a sync node with key slot = OID(OSID1 only). 1 = Sync frame received on channel A / node configured to transmit sync frames; 0 = No sync frame received on channel A / node not configured to transmit sync frames</p> <p>Description FRF.FID, modified: Determines the frame ID to be rejected by the FIFO. With the additional configuration of register FRFM, the corresponding frame ID filter bits are ignored, which results in further rejected frame IDs. When FRFM.MFID is zero, a frame ID filter value of zero means that no frame ID is rejected.</p> <p>Description MHDF.WAHP, added: Outside DEFAULT_CONFIG and CONFIG state this flag is set by the CC when the message handler tries to write message data into the header partition of the Message RAM due to faulty configuration of a message buffer...</p> <p>Description MONITOR_MODE, completed: ...In this mode the CC is able to receive FlexRay frames and to detect wakeup pattern...</p> <p>Modified the description of CUST1.POBFEN to: Parity Error Reporting of Output Buffer (OBF1/OBF1) RAMs Enable/Test Disable; 0_B: Parity error in Output Buffer (OBF1,OBF2) RAMs is not reported to SCU nor to OBF1, OBF2 RAM wrapper. 1_B: The parity error reporting and thus the activation of error signals to SCU as also to OBF1 and OBF2 RAM wrapper is enabled.</p> <p>Modified the description of CUST1.PMBFEN to: Parity Error Reporting of Message Buffer (MBF) RAMs Enable/Test Disable; 0_B: Parity error in Message Buffer (MBF) RAMs is not reported to SCU nor to MBF RAM wrapper. 1_B: The parity error reporting and thus the activation of error signals to SCU as also to MBF RAM wrapper is enabled.</p> <p>Modified the description of CUST1.PITBFEN to: Parity Error Reporting of Message Buffer (IBF1, IBF2, TBF1, TBF2) RAMs Enable/Test Disable; 0_B: Parity error in Input Buffer and Transient Buffer (IBF1, IBF2, TBF1, TBF2) RAMs is not reported to SCU nor to IBF1, IBF2, TBF1, and TBF2 RAM wrapper. 1_B: The parity error reporting and thus the activation of error signals to SCU as also to IBF1, IBF2, TBF1, and TBF2 RAM wrapper is enabled</p>

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_2.0	<p>Modified the address of INT0SRC, INT1STC, TINT0SRC, TINT1SRC, ERAY_NDIC1...ERAY_NDIC4, ERAY_MSIC1...ERAY_MSIC4, NDAT_NDAT0SRC, ERAY_NDAT1SRC, ERAY_MBSC0SRC, ERAY_MBSC1SRC.</p> <p>Cleared up the Index entries.</p> <p>Changed Address Range from to 00000000_H ...00007FFF_H to F0010000_H ...F0017FFF_H</p>
Rev_2.1	<p>Inserted additional information on the minimal eray_bclk required to work properly.</p> <p>Corrected some Typos in the port implementation.</p> <p>Renamed the Interrupt Control Registers due to some problems extracting register names by automatic tools.</p> <p>Changed in Figure 2-1 the names of the clock signals.</p> <p>Changed the Long Name of Register ACS to Aggregated Channel Status Register.</p> <p>Included Message Handler Timing Details.</p> <p>Modified the Clock Signal Names in Figure 2-37.</p>
Rev_2.2	<p>Modified the Pinning of TC1797: TXDA: P0.14, RXDA:P0.9; TXENA: P0.10; TXDB: P0.12; RXDB: P0.12; TXENB: P0.11. Changed the name of the Bit RFCLL in SILS Register.</p>
Rev_2.3	<p>Included IP Specification changes of Revision 1.2.3.</p> <p>Description write access to data section of IBF, changed: ...If not all bytes of a 32-bit word have been written by the Host (8/16-bit access only), WRDSn holds partly undefined data.</p> <p>Description SIR.WST, completed: This flag is set when the wakeup status vector CCSV.WSV is changed by a protocol event.</p> <p>Description SIR.SWE, modified: This flag is set after a stop watch activation when the actual cycle counter and macrotick value are stored in the Stop Watch register (see section Stop Watch Register 1 (STPW1)).</p> <p>Description STPW1.ESWT, corrected: ...In single-shot mode this bit is reset to 0 after the actual cycle counter and macrotick value are stored in the Stop Watch register.</p> <p>Description SUCC1.CMD3:0, changed: RESET_STATUS_INDICATORS Resets status flags CCSV.CSNI, CCSV.CSAI, and CCSV.WSV to their default values. May be called in POC state "READY". When called in any other state, CMD will be reset to 0000_B = command_not_accepted.</p> <p>Description CCSV.POC, added: ...101011_B = STARTUP_SUCCESS state 101100_B ... 111111_B = reserved</p>

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_2.3	<p>Description SUCC1.MTSA,B, added: Note: MTSA,B may also be changed outside “DEFAULT_CONFIG” or “CONFIG” state when the write to SUCC1 register is directly preceded by the unlock sequence as described in chapter Lock Register (LCK). This may be combined with CHI command SEND_MTS...</p> <p>Description Communication Controller Status Registers, added: ...The status vector may change faster than the Host can poll the status vector, depending on eray_bclk frequency.</p> <p>Description Communication Controller Status Registers, removed: ...All internal counters and the Communication Controller status flags are reset when the Communication Controller transits from “CONFIG” to “READY” state.</p> <p>Description CCSV.FSI, changed: ...Reset by transition from HALT to “DEFAULT_CONFIG” state.</p> <p>Description CCSV.HRQ, changed: ...Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.</p> <p>Description CCSV.SLM, changed: ...Set to the value defined by SUCC1.TSM when entering “READY”, “CONFIG”, or “DEFAULT_CONFIG” state.</p> <p>Description CCSV.CSI, completed: ...The flag is set whenever the POC enters READY state due to CHI command READY.</p> <p>Description CCSV.WSV, modified: ...000_B = UNDEFINED. Wakeup not yet executed by the Communication Controller...</p> <p>Description CCSV.RCA, corrected: ...The “RUN” command resets this counter to the maximum number of coldstart attempts as configured by SUCC1.CSA.</p> <p>Description Communication Controller Status Registers, removed: ...Note: CHI command “RESET_STATUS_INDICATORS” (SUCC1.CMD = 1010_B) resets flags FSI, HRQ, CSNI, CSAI, the slot mode SLM, and the wakeup status WSV.</p> <p>Description CC Error Vector, changed: Reset by transition from “HALT” to “DEFAULT_CONFIG” state or when entering “READY” state...</p> <p>Description MONITOR_MODE, added: ...In “MONITOR_MODE” the Communication Controller is not able to distinguish between CAS and MTS symbols. In case one of these symbols is received on one or both of the two channels, the flags SIR.MTSA resp. SIR.MTSB are set. SIR.CAS has no function in “MONITOR_MODE”.</p> <p>Inserted remark to negate eray_ibusy and eray_obusy signal to simplify the usage for DMA requesting and interrupt triggering.</p> <p>Corrected Typo in CMD field of SUCC1.</p>

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_2.4	Updated ERAY Kernel Errata List (Bosch Errata List from November 24th, 2006: "Errata_E-Ray_R1.0_20061124")
Rev_2.5	<p>Modified RISA: Receive Input Select Channel A</p> <p>00_B Channel A receiver input RXDA0 selected 01_B Channel A receiver input RXDA1 selected 10_B Channel A receiver input RXDA2 selected 11_B Channel A receiver input RXDA3 selected</p> <p>Modified RISB: Receive Input Select Channel B</p> <p>00_B Channel B receiver input RXDB0 selected 01_B Channel B receiver input RXDB1 selected 10_B Channel B receiver input RXDB2 selected 11_B Channel B receiver input RXDB3 selected</p> <p>Description MONITOR_MODE, removed: ...In "MONITOR_MODE" the receive FIFO is not available.</p> <p>Description Message RAM Configuration Register, added:...Note:... In case two or more message buffers are assigned to slot 1 by use of cycle filtering, all of them must be located either in the "Static Buffers" or at the beginning of the "Static + Dynamic Buffers" section...</p> <p>Description CCSV.POC[5:0], added: ...101011_B = "STARTUP_SUCCESS" state 101100_B...111111_B = reserved.</p> <p>Inserted text condition to commonly cover IP revision 1.0.0 and IP revision 1.0.1.</p> <p>Updated ERAY Kernel Errata List (Bosch Errata List from December 20th, 2006: "Errata_E-Ray_R1.0.0_20061220").</p> <p>Included IP Specification changes of Revision 1.2.5.</p> <p>Description EIR.SFO, changed: Set when either the number of sync frames received during the last communication cycle or the total number of sync frames received during the last double cycle exceeds the maximum number of sync frames as defined by GTUC2.SNM.</p> <p>Description SUCC1.CMD[3:0], changed: RESET_STATUS_INDICATORS ... May be called in POC states "READY" and "STARTUP"...</p>

Table 2-28 Revision History (cont'd)

Version Number	Changes to Previous Version
Rev_2.5	<p>Description CCSV.SLM, changed: Indicates the actual slot mode of the POC in states "READY", "STARTUP", "NORMAL_ACTIVE", and "NORMAL_PASSIVE". Default is SINGLE. Changes to ALL, depending on SUCC1.TSM. In "NORMAL_ACTIVE" or "NORMAL_PASSIVE" state the CHI command ALL_SLOTS will change the slot mode from SINGLE over ALL_PENDING to ALL. Set to SINGLE in all other states.</p> <p>Description SFS.VSAE, removed: ... (vSyncFramesEvenA)</p> <p>Description SFS.VSAO, removed: ... (vSyncFramesOddA)</p> <p>Description SFS.VSBE[3:0], removed:... (vSyncFramesEvenB)</p> <p>Description SFS.VSBO[3:0], removed: ... (vSyncFramesOddB)</p> <p>Description MBS.RCIS, MBS.SFIS, MBS.SYNS, MBS.NFIS, MBS.PPIS, MBS.RESS, completed: For receive buffers (CFG = 0) the following status bits are updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flags have no meaning and should be ignored.</p> <p>Description Network Management, added: Note: ...When the Communication Controller transits to "HALT" state, the cycle count is not incremented and therefore the NM vector is not updated. In this case NMV1...3 holds the value from the cycle before.</p> <p>Description Configuration of the FIFO, added: Note: It is recommended to program the MBI bits of the message buffers belonging to the FIFO to 0 via WRHS1.MBI to avoid generation of RX interrupts...</p> <p>Updated the ERAY_CLC reset value to 0000 0100_H.</p> <p>Included the ERAY trigger signals to the DMA channels 10 -17.</p>

3 System Control Unit (SCU)

The SCU controls all system relevant tasks independent if they are needed for Normal Mode or for JTAG Enabled Mode only. The system tasks of the SCU are:

- Reset operation (see [Chapter 3.2](#))
- System clock control (see [Chapter 3.3](#))
- Power supply system (see [Chapter 3.4](#))
- System Interrupt control (see [Chapter 3.5](#))

3.1 SCU Registers

Figure 3-1 shows all registers associated with the SCU.

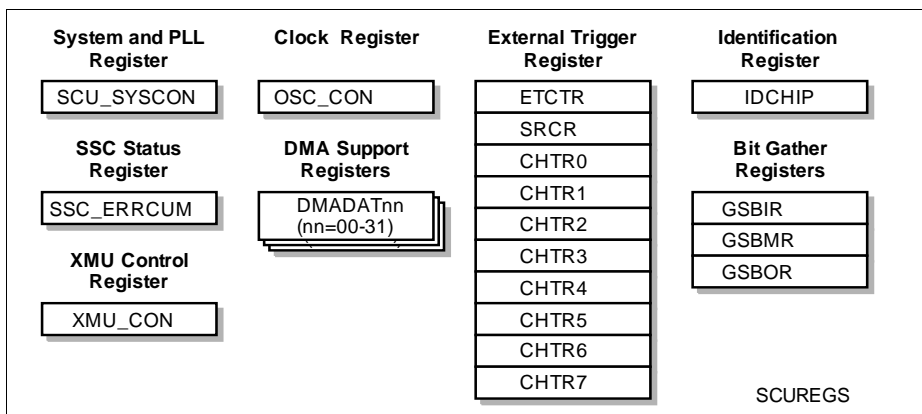


Figure 3-1 SCU Registers

Table 3-1 SCU Kernel Register Address Space

Module	Base Address	End Address	Note
SCU	0000 0800 _H	0000 08FF _H	256 byte

Table 3-2 SCU Registers

Register Short Name	Register Long Name	Address Offset	Description see
OSCCON	Oscillator Control Register	0000 0000 _H	Page 3-11
-	Reserved	0000 0004 _H	-
-	Reserved	0000 0008 _H	-

System Control Unit (SCU)
Table 3-2 SCU Registers (cont'd)

Register Short Name	Register Long Name	Address Offset	Description see
-	Reserved	0000 000C _H	-
-	Reserved	0000 0010 _H	-
-	Reserved	0000 0014 _H	-
-	Reserved	0000 0018 _H	-
-	Reserved	0000 001C _H	-
SYSCON	System and PLL Control Register	0000 0020 _H	Page 3-24
CHTR0	Channel 0 Trigger Register	0000 0030 _H	Page 3-45
CHTR1	Channel 1 Trigger Register	0000 0034 _H	Page 3-45
CHTR2	Channel 2 Trigger Register	0000 0038 _H	Page 3-45
CHTR3	Channel 3 Trigger Register	0000 003C _H	Page 3-45
CHTR4	Channel 4 Trigger Register	0000 0040 _H	Page 3-45
CHTR5	Channel 5 Trigger Register	0000 0044 _H	Page 3-45
CHTR6	Channel 6 Trigger Register	0000 0048 _H	Page 3-45
CHTR7	Channel 7 Trigger Register	0000 004C _H	Page 3-45
ETCTR	External Trigger Control Register	0000 0050 _H	Page 3-36
-	Reserved	0000 0054 _H	-
SRCR	Service Request Control Register	0000 0058 _H	Page 3-41
SSC_ ERRCUM	SSC Cumulative Error Register	0000 005C _H	Page 3-31
IDCHIP	Chip Identification Register	0000 0060 _H	Page 3-33
XMU_CON	XMU Global Control Register	0000 0064 _H	Page 3-47
GSBIR	Gather Scattered Bits Input Register	0000 0068 _H	Page 3-52
-	Reserved	0000 006C _H	-
GSBMR	Gather Scattered Bits Mask Register	0000 0070 _H	Page 3-53
-	Reserved	0000 0074 _H	-
GSBOR	Gather Scattered Bits Output Register	0000 0078 _H	Page 3-54
-	Reserved	0000 007C _H	-
DMADAT00	DMA Data 00 Register	0000 0080 _H	Page 3-50
DMADAT01	DMA Data 01 Register	0000 0084 _H	Page 3-50
DMADAT02	DMA Data 02 Register	0000 0088 _H	Page 3-50

System Control Unit (SCU)

Table 3-2 SCU Registers (cont'd)

Register Short Name	Register Long Name	Address Offset	Description see
DMADAT03	DMA Data 03 Register	0000 008C _H	Page 3-50
DMADAT04	DMA Data 04 Register	0000 0090 _H	Page 3-50
DMADAT05	DMA Data 05 Register	0000 0094 _H	Page 3-50
DMADAT06	DMA Data 06 Register	0000 0098 _H	Page 3-50
DMADAT07	DMA Data 07 Register	0000 009C _H	Page 3-50
DMADAT08	DMA Data 08 Register	0000 00A0 _H	Page 3-50
DMADAT09	DMA Data 09 Register	0000 00A4 _H	Page 3-50
DMADAT10	DMA Data 10 Register	0000 00A8 _H	Page 3-50
DMADAT11	DMA Data 11 Register	0000 00AC _H	Page 3-50
DMADAT12	DMA Data 12 Register	0000 00B0 _H	Page 3-50
DMADAT13	DMA Data 13 Register	0000 00B4 _H	Page 3-50
DMADAT14	DMA Data 14 Register	0000 00B8 _H	Page 3-50
DMADAT15	DMA Data 15 Register	0000 00BC _H	Page 3-50
DMADAT16	DMA Data 16 Register	0000 00C0 _H	Page 3-50
DMADAT17	DMA Data 17 Register	0000 00C4 _H	Page 3-50
DMADAT18	DMA Data 18 Register	0000 00C8 _H	Page 3-50
DMADAT19	DMA Data 19 Register	0000 00CC _H	Page 3-50
DMADAT20	DMA Data 20 Register	0000 00D0 _H	Page 3-50
DMADAT21	DMA Data 21 Register	0000 00D4 _H	Page 3-50
DMADAT22	DMA Data 22 Register	0000 00D8 _H	Page 3-50
DMADAT23	DMA Data 23 Register	0000 00DC _H	Page 3-50
DMADAT24	DMA Data 24 Register	0000 00E0 _H	Page 3-50
DMADAT25	DMA Data 25 Register	0000 00E4 _H	Page 3-50
DMADAT26	DMA Data 26 Register	0000 00E8 _H	Page 3-50
DMADAT27	DMA Data 27 Register	0000 00EC _H	Page 3-50
DMADAT28	DMA Data 28 Register	0000 00F0 _H	Page 3-50
DMADAT29	DMA Data 29 Register	0000 00F4 _H	Page 3-50
DMADAT30	DMA Data 30 Register	0000 00F8 _H	Page 3-50
DMADAT31	DMA Data 31 Register	0000 00FC _H	Page 3-50

3.2 Reset Control Block

The single system reset function initializes the SAK-CIC310-OSMX2HT into a defined default state and is invoked by an external $\overline{\text{PORST}}$ reset (Power-on reset) indicated by hardware reset input after power-on.

All reset are asynchronous resets, therefore the reset trigger and activation is asynchronous to the system clock and does not rely on it. The deactivation or termination is synchronous to the system clock for the $\overline{\text{PORST}}$.

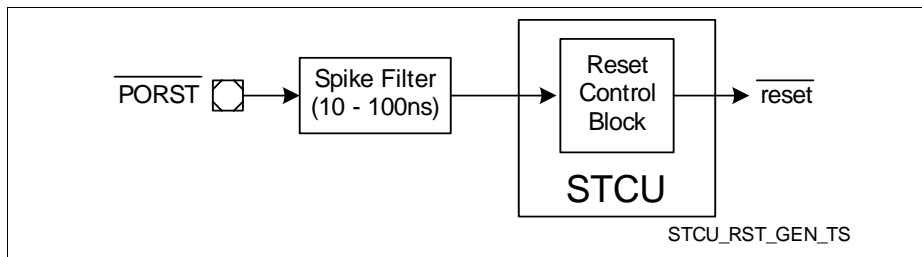


Figure 3-2 Reset Logic

Power-On Reset

The $\overline{\text{PORST}}$ input pin requests a power-on reset. Driving the $\overline{\text{PORST}}$ pin low causes a non-synchronized reset of the entire device. After releasing the $\overline{\text{PORST}}$ input the device continues with the boot (startup) sequence after the internal extension of the reset sequence.

Note: The trailing edge for the $\overline{\text{PORST}}$ pin activation should not need longer than 50 ms. The $\overline{\text{PORST}}$ pin itself can be asserted longer than 50 ms

$\overline{\text{PORST}}$ is equipped with a noise suppression filter which suppresses glitches below 10 ns pulse width. $\overline{\text{PORST}}$ pulses with a width above 100 ns are safely recognized.

To enable a clean power-up, the majority of pins have a “enable pad supply” (ENPS) function. This ENPS is pad supply (V_{DDP}) driven. So only if the pad supply is stable, the output pads are activated. Five pins do not support ENPS: JTAGEN, $\overline{\text{PORST}}$, XTAL1, and XTAL2. In case of a low voltage signal at the input port ENPS the output port PAD activates the weak pull-up and disables the output driver independent of the port direction. In case of a high voltage signal at the input port ENPS the bidirectional platform pads operate in normal mode.

3.2.1 Reset Length

Inside the reset state machine a counter RSTCNT for the control of the reset length is implemented.

System Control Unit (SCU)

The falling edge of a reset source (e.g. the power-on reset) is activating an internal reset state machine and forces the reset counter RSTCNT to the specified reset value. An additional falling edge does not reactivate the reset state machine if already active.

Table 3-3 Reset Values of RSTCNT

Reset Source	RSTCNT Start Value
Power-On Reset	8
SW Reset	8

As soon as the reset request is inactive, the reset counter decrements on every clock pulse. If $RSTCNT > 0$, the internal reset signal is held active. When reaching 0, the internal reset signal is deasserted only if the reset request is also deasserted and the device can continue with the defined non reset operation.

3.3 Clock System

3.3.1 Overview

This chapter describes the SAK-CIC310-OSMX2HT clock system. Topics covered include clock generation, clock domains, the operation of clock circuitry, power-up operation for the clock system, fail-safe operation, and clock control registers.

The SAK-CIC310-OSMX2HT clock system performs the following functions:

- Uses internal free running frequency of the VCO block to create a fast clock frequency f_{CGU} .
- Acquires and buffers external clock signal (XTAL1 and/or XTAL2) to create a fast clock frequency f_{CGU} .
- Distributes in-phase synchronized clock signal throughout the SAK-CIC310-OSMX2HT's entire clock tree.

The clock system must be operational before the SAK-CIC310-OSMX2HT can operate, therefore containing special logic to handle power-up, reset, and fault handling operations.

3.3.2 Clock Generation Unit

The Clock Generation Unit (CGU) allows a very flexible clock generation for SAK-CIC310-OSMX2HT.

Features

The Clock Generation Unit serves different purposes:

- PLL Feature for multiplying clock source by different factors
- Direct Drive for direct clock put through

System Control Unit (SCU)

- Comfortable state machine for secure switching between basic PLL, direct or prescaler operation
- Sleep and Power Down Mode support

Note: Not all features are supported in every product.

The Clock Generation Unit (CGU) in the SAK-CIC310-OSMX2HT consists of an oscillator circuit (COSC), a Phase-Locked Loop (PLL) module, and a Clock Multiplexing Unit (CMU). The PLL can convert a low-frequency external clock signal from the oscillator circuit to a high-speed internal clock. In general, the clock generation unit is controlled through the System Control Unit (SCU) module of the SAK-CIC310-OSMX2HT.

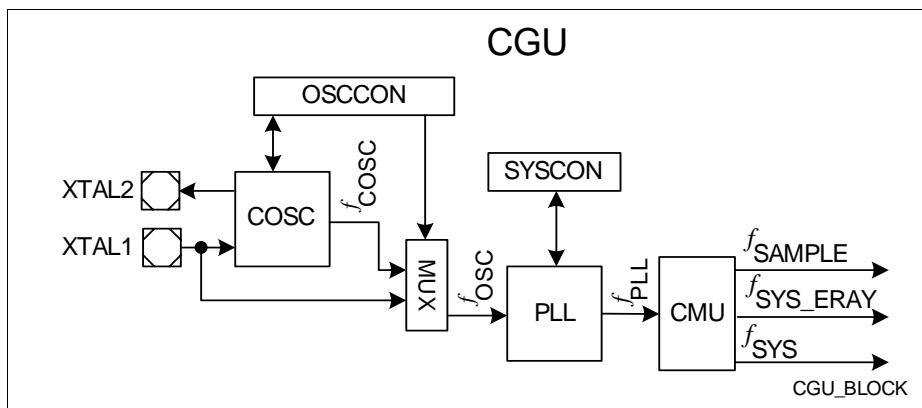


Figure 3-3 Clock Generation Unit Block Diagram

The following sections describe the different parts of the CGU:

3.3.2.1 Crystal Oscillator Circuit (COSC)

The crystal oscillator circuit, designed to work with both, an external crystal oscillator or an external stable clock source, basically consists of an inverting amplifier with XTAL1 as input, and XTAL2 as output.

When using a crystal, a proper external oscillator circuitry must be used, connected to both pins, XTAL1 and XTAL2. The on-chip oscillator frequency can be within the range of 20 MHz or 40 MHz for the new low Jitter PLL.

The size of the capacitors is very much dependant from the crystal used and has to be optimized individually.

When using an external clock signal it must be connected to XTAL1 and XTAL2 is left open (unconnected).

Figure 3-5 shows the recommended external circuitries for both operating modes, External Crystal Mode and External Input Clock Mode.

External Crystal Mode

When using an external crystal, its frequency can be alternatively 20 MHz or 40 MHz. An external oscillator load circuitry must be used, connected to both pins, XTAL1 and XTAL2. It consists normally of the two load capacitances C1 and C2, for some crystals a series damp resistor might be necessary. The exact values and related operating range are dependent on the crystal and have to be determined and optimized together with the crystal vendor using the negative resistance method.

External load capacitances have to be connected to V_{SSPOSC} for SAK-CIC310-OSMX2HT. Further information regarding correct IEMC/EMV layout of the PCB can be found in Infineon EMC/EMV guidelines, available via download under: [Link to the PDF:](http://www.infineon.com/upload/Document/cmc_upload/documents/009/712/ap242005.pdf)
http://www.infineon.com/upload/Document/cmc_upload/documents/009/712/ap242005.pdf.

The General Microcontroller EMC Guideline is available via download: [Link to the PDF:](http://www.infineon.com/upload/Document/AP2402631_EMC_Guidelines.pdf)
http://www.infineon.com/upload/Document/AP2402631_EMC_Guidelines.pdf

Special EMC Guideline for the TC1796 can be downloaded from: [Link to the PDF:](http://www.infineon.com/upload/Document/ap3208610_TC1796_EMC_Design_Guideline.pdf)
http://www.infineon.com/upload/Document/ap3208610_TC1796_EMC_Design_Guideline.pdf

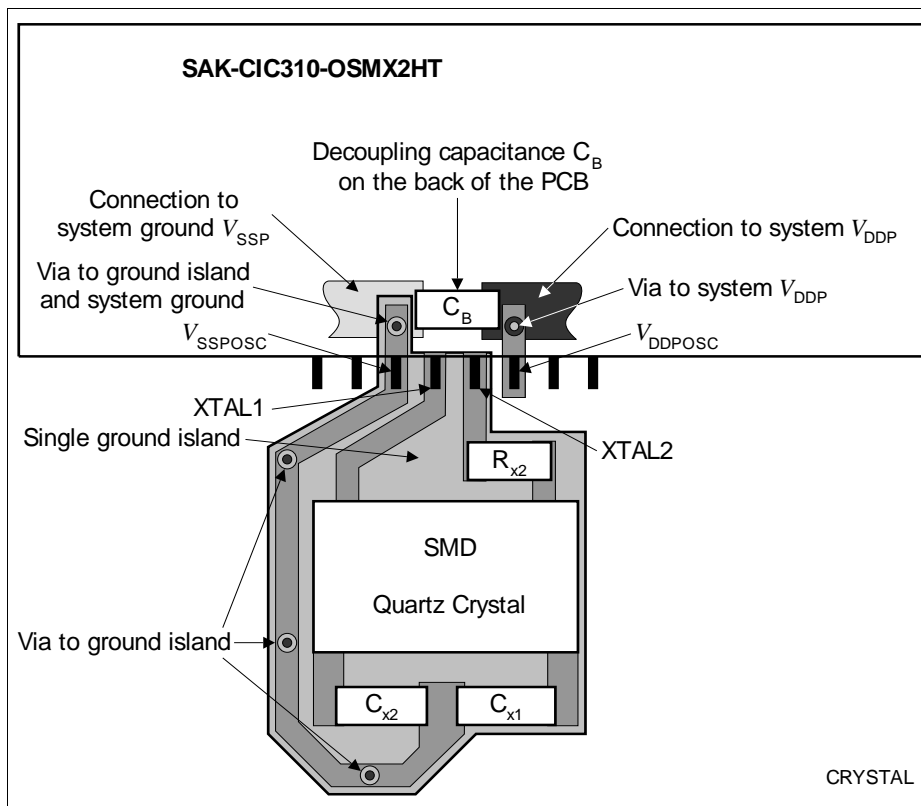


Figure 3-4 Layout Example for a SMD Quartz Crystal

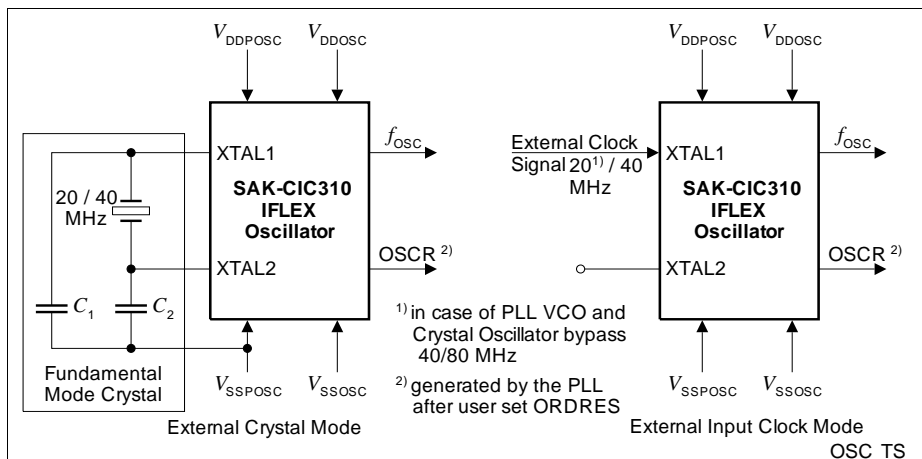


Figure 3-5 SAK-CIC310-OSMX2HT External Circuitry for the Internal Oscillator

Oscillator JTAG Enabled Mode Configuration

To allow device test without distortion by the oscillator circuitry, the following mode is available:

The crystal oscillator circuitry is disabled and f_{osc} is directly derived from XTAL1 (**OSCCON.OSCBY** = 1).

In normal operating mode the oscillator is running and f_{osc} is derived from the crystal or from an external clock signal (**OSCCON.OSCBY** = 0).

The **OSCCON.OSCBY** bit can be set:

- Writing to the register
- During active \overline{PORST} to pull \overline{BYPASS} to ground.

Oscillator Run Detection

To detect during start-up after power-on, if the oscillator is running or not, an oscillator run detection logic is included. By this it is possible to drive an error signal after the oscillator run condition (**OSCCON.OSCR** = 0) is met. This has to be triggered by resetting and starting the oscillator run detection logic (**OSCCON.ORDRES**=1) within the application software.

The oscillator run detection monitors the incoming clock from the oscillator and checks if it fits for an operation in Normal Mode with the selected setting of the N-Divider. Only incoming frequencies that are too low to enable a stable operation of the VCO circuit are detected.

System Control Unit (SCU)

The oscillator run detection consists of two counters. Counter A runs at the oscillator frequency and stops on reaching its maximum count of 2^m . Counter B runs at a frequency derived from the PLL. When counter B reaches its maximum count, (2^n), the state of counter A is latched into a flip-flop C and both counters are reset. Before any Internal_RESET has occurred, the state of the signal `osc_running` is invalid. After the first maximum count of counter B the Internal_RESET signal resets both counter A and counter B. At the second maximum count of counter B, the flip-flop C takes the state of counter A and becomes defined. The circuit can start without an external reset and becomes defined after at least 2×2^n input pulses at counter B.

For proper detection of the oscillator operation the time-out of counter B ($2^n \times (1/f_{div})$) must be longer than the time-out of counter A ($2^m \times (1/f_{osc})$).

As the frequency of counter B is the frequency of the feedback divider of the PLL, m has to be smaller than n . In the locked state holds: $f_{osc}/P = f_{div}$. For detecting a proper oscillator frequency, the time $2^n \times (N/f_{vco_free})$ must be longer than $2^m \times (1/f_{osc})$.

The current values for m is 3 (divide by 8), for n is 5 (divide by 32). Thus for the minimum frequency of $f_{osc} = 20$ MHz, the maximum frequency allowed on f_{div} is 80 MHz. This means for $f_{vco_free} = 400$ MHz and $N=12$, this condition is always met. The circuit is initialized after a minimum of $2\mu s$, maximum $11.7\mu s$.

Via the bit ORDRES in the PLL_CON register the detection can be restarted also during normal operation or e.g. in case of loss-of-lock condition.

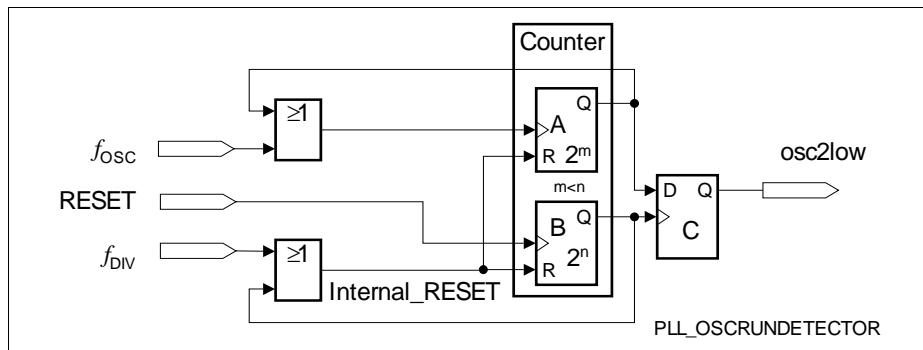


Figure 3-6 Oscillator Run Detection

OSCCON

Oscillator Control Register

(00_H)

Reset Value: 0000 000C_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										OSC BY	OSC DISC	OGC	ORD RES	OSC R	
r										rw	rwh	rw	rwh	rh	

Field	Bits	Type	Description
OSCR	0	rh	Oscillator Run Status Bit This bit shows the state of the oscillator run state. 0 The oscillator is not running. 1 The oscillator is running. This bit is updated by signal osc2low from the PLL.
ORDRES	1	rwh	Oscillator Run Detection Reset 0 No operation (default) 1 The oscillator run detection logic is reset and restarted. This bit will be automatically be cleared after it was set. After Reset the Oscillator Run Detection is deactivated and has to be activated by the application software.
OGC	[3:2]	rw	Oscillator Gain Control Only the 11 _B setting should be used. Anyway the other setting can be used for test and characterization purposes. Default Value: 11 _B
OSCDISC	4	rwh	Oscillator Disconnect 0 Oscillator is connected to the PLL (default) 1 Oscillator is disconnected from the PLL

System Control Unit (SCU)

Field	Bits	Type	Description
OSCBY	5	rw	Crystal Oscillator Bypass 0 Normal operation, crystal oscillator is not bypassed 1 Crystal Oscillator is bypassed This bit has only an effect if the crystal oscillator is used, otherwise it is ignored.
0	[31:6]	r	Reserved; read as 0; should be written with 0.

The following values/signals are synchronized internally to their corresponding clocks. This avoids unexpected glitches on the output:

- **SYSCON.KDIV** with f_{PLL} : The clock period of f_{PLL} changes after maximum 5 cycles with a rising edge of f_{PLL} .
- **SYSCON.PLLCTRL** with f_{VCO}/f_P ($f_P = f_{OSC}/P$): The clock period of f_{PLL} changes after maximum 7 cycles with a rising edge of f_{PLL} .

All the other signals (e.g. **OSCCON.OSCDISC**, **SYSCON.NDIV**, **SYSCON.PLLVB**) are not synchronized, so any change on these signals may cause glitches on the output clock. Therefore these values must only be changed while the PLL is bypassed, e.g. **SYSCON.PLLCTRL** = 01_B.

The system clock f_{CGU} (for Normal Mode equal to f_{PLL}) generated from an oscillator clock f_{OSC} in one of four selectable ways:

- Bypass Mode (Direct Drive)
- Prescaler Mode
- Normal Mode (PLL Mode)
- Free running Mode (PLL Base Mode)

Bypass Mode

When Bypass Mode is configured the SAK-CIC310-OSMX2HT clock system is directly fed from the external clock input, i.e. $f_{sys} = f_{OSC}$. This allows operation of the SAK-CIC310-OSMX2HT with a reasonably small fundamental mode crystal. The specified minimum values for the system clock phases must be respected. Therefore the maximum input clock frequency depends not only on the cycle time, but also on the clock signal's duty cycle.

Please note, that it is not possible to switch from direct drive to the other modes nor the other way round.

Prescaler Mode

When prescaler operation is configured the SAK-CIC310-OSMX2HT input clock f_{OSC} is divided down for low power operation.

Normal Mode

The Phase-Locked Loop (PLL) converts a low-frequency external clock signal to a high-speed internal clock for maximum performance. The system clock f_{sys} signal is generated by the PLL module.

Free Running Mode

The Phase-Locked Loop (PLL) is not locked to an input clock and the device clocks are generated out of the VCO base frequency.

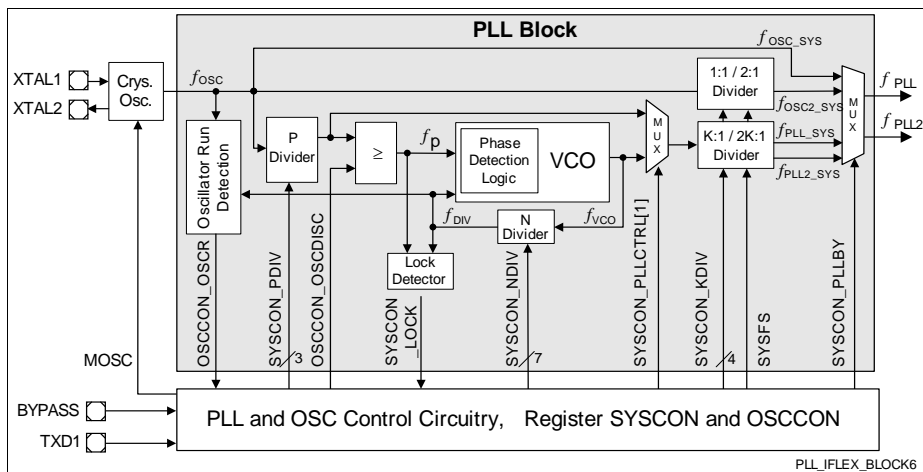


Figure 3-7 CGU Detailed Block Diagram

3.3.3 Phase-Locked Loop (PLL) Module

This section describes the SAK-CIC310-OSMX2HT PLL module. The PLL is a main component of CGU part dedicated for generating the system clock. The PLL converts a low-frequency external clock signal to two high-speed internal clocks for maximum performance.

Features

- Programmable clock generation PLL
- On chip Loop filter
- Fully differential analog circuitry
- Input frequency: $f_{\text{OSC}} = 20$ to 40 MHz (P fixed to 1)
- VCO frequency: $f_{\text{VCO}} = 400$ to 500 MHz (one range only)
- Output frequency: $f_{\text{PLL}} = 15$ to 80 MHz
- VCO lock detection
- Oscillator run detection
- 4bit input divider **P**: (divide by PDIV+1)
- 8bit feedback divider **N**: (multiply by NDIV+1, stability restrictions may apply)
- 5bit output divider **K**: (divide by KDIV+1)
- Synthesize Mode
- Bypass Mode
- Prescaler Mode
- Freerunning Mode
- Sleep Mode
- Glitchless programming of output divider K.
- Glitchless switching between Normal Mode and Prescaler Mode
- Glitchless programming of output divider K
- During normal operation all control signals may be changed randomly and at any time. Only if modifying the output divider **K** some restriction exist (maximum change rate).

Note: To achieve the low jitter required of the FlexRay Communication controller, the P, N, and K counters have only a very limited programmability within the SAK-CIC310-OSMX2HT.

Ignoring the P divider; this defines the Freerunning Mode

Sleep Mode

Sleep Mode and Power Down Mode is not supported. Only a VCO sleep mode is supported by setting **SYSCON.PLLCTRL** = 01_B.

Normal Mode

In Normal Mode the input clock f_{OSC} is divided down by a factor P, multiplied by a factor N and then divided down by a factor K.

So the output frequency is given by:

$$f_{PLL} = \frac{N}{P \cdot K} \cdot f_{OSC} = \frac{(NDIV + 1)}{(PDIV + 1) \cdot (K2DIV + 1)} \cdot f_{OSC} \quad (3.2)$$

The Normal Mode is selected by setting **SYSCON.PLLCTRL** = 11_B.

PLL Bypass Mode

In Bypass Mode the input clock f_{OSC} is directly connected to the PLL output f_{CGU} and f_{CGU2} . With PLL2SEL the frequency of f_{PLL2} can be controlled as in normal mode, but changes in f_{PLL2} are asynchronous. The state of f_{PLL2} is undefined (0 or 1) after power-up when PLL2_en=1 and PLL2SEL=0 but the output frequency is $f_{PLL2}/2$. When $f_{osc}=0$ and PLL2_en=1 a 1 pulse at PLL2SEL initializes f_{PLL2} to 0.

In normal bypass mode the analog block and the lock detection are still running. The lock signal shows the actual state of the control loop.

So the output frequency is given by:

$$f_{CGU} = f_{OSC} \quad (3.3)$$

The Bypass Mode is selected by setting **SYSCON.PLLBY** = 1_B.

Please note, that it is not possible to switch from direct drive to the other modes nor the other way round due to possible clock glitches generated.

Prescaler Mode

In this mode the input clock f_{OSC} is divided by the output divider K for f_{PLL} and K or 2*K for f_{PLL2} .

In normal prescaler mode the analog block and lock detection keeps on running. The lock signal shows the actual state of the control loop.

System Control Unit (SCU)

If the PLL input clock is disabled, the PLL enters the free running mode. The PLL outputs are forced asynchronous to the actual VCO frequency divided by the K factor. The actual VCO frequency depends on the history.

If the VCO is put in power-down during prescaler mode, the analog block and lock detection is powered down. In this mode lock is always 0.

If the PLL input clock is disabled in this power save prescaler state, no PLL clock is available.

In Prescaler Mode the input clock f_{OSC} is divided down by a factor $P \times K$.

So the output frequency is given by

$$f_{CGU} = \frac{f_{OSC}}{(P \cdot K)} = \frac{f_{OSC}}{((PDIV + 1) \times (K2DIV + 1))} \quad (3.4)$$

The Prescaler Mode is selected by setting **SYSCON.PLLCTRL** = 0X_B.

Freerunning Mode

For a lot of chips an oscillator watchdog (external to the PLL module) is available that disables the PLL input in the case of an UNLOCK condition, e.g. in the case of a broken crystal or too high an input clock frequency.

In Freerunning Mode the base frequency output of the Voltage Controlled Oscillator (VCO) $f_{VCObase}$ is only divided down by a factor K (KDIV).

So the output frequency is given by

$$f_{CGU} = \frac{f_{VCObase}}{K} = \frac{f_{VCObase}}{(K2DIV + 1)} \quad (3.5)$$

The Freerunning Mode is selected by setting **SYSCON.PLLCTRL** = 10_B. If this signal is set (or if the input clock is missing) the phase detector pulls the VCO down to the lower limit frequency, i.e. the free-running frequency f_{free} .

General Configuration Overview

All three divider values and all necessary other values can be configured via the PLL configuration register **SYSCON**.

Table 3-4 shows a only possible values for the P factor and gives the valid output frequency range for the P divider dependent on the f_{OSC} frequency range:

Table 3-4 P-Divider Factors

P = PDIV+1	PDIV	f_{REF} for $f_{OSC} =$	
		20 MHz	40 MHz
1	0	20	40

Note: Of course the whole range in between two f_{OSC} columns in the above table is allowed. E.g. for a range $f_{OSC} = 20$ to 25 , and $P = 3$, $f_{REF} = 6.66$ to 8.33 MHz.

The P-divider output frequency f_{REF} is fed to a Voltage Controlled Oscillator (VCO). The VCO is a part of PLL with a feedback path. A divider in the feedback path (N divider) divides the VCO frequency. Additionally to N the correct range of f_{VCO} must be chosen by configuring **SYSICON.PLLVB**:

:

Table 3-5 VCO Ranges

VCOSSEL [1:0]	f_{VCOmin}	f_{VCOMax}	$f_{VCObase}^{1)}$	$f_{oscmin}^{2)}$	Unit
00	400	500	approx. 150-320	1.5	MHz

1) $f_{VCObase}$ is the free running operation frequency of the PLL, when no input clock is available

2) Minimum oscillator frequency to allow oscillator run detection to work properly.

The VCO band (400...500 MHz, 500...600 MHz, 600...700 MHz, 700...800 MHz) must be selected according to the desired VCO output frequency. The following graphic illustrates how this output frequency depends on the input frequency and the multiplication factor.

The following table shows the possible N loop division rates and gives the valid output frequency range for f_{REF} depending on N and the VCO frequency range:

Table 3-6 N Loop Division Rates

N = NDIV+1	NDIV	f_{REF} for $f_{VCO} =$	
		400	500
≤ 11	≤ 10	not allowed	
12	11	36,36	45,45
13...19	12...18	not allowed	
20	19	20,00	25,00
21	20	19,05	23,81
22	21	18,18	22,73

Table 3-6 N Loop Division Rates (cont'd)

N = NDIV+1	NDIV	f_{REF} for $f_{VCO} =$	
		400	500
...			
98	97	4,08	5,10
99	98	4,04	5,05
100	99	4,00	5,00
≥ 101	≥ 100	not allowed	

Note: Of course the whole range in between two f_{VCO} columns in the above table is allowed.

The N-divider output frequency f_{DIV} is then compared with f_{REF} in the phase detector logic, which is within the VCO logic. The phase detector determines the difference between the two clock signals and accordingly controls the output frequency of the VCO, f_{VCO} .

Note: Due to this operation, the VCO clock of the PLL has a frequency which is a multiple of f_{REF} . The factor for this is controlled through the value applied to the N-divider in the feedback path. For this reason this factor is often called a multiplier, although it actually controls division.

System Control Unit (SCU)

The output frequency of the VCO, f_{VCO} , is divided by K to provide the final desired output frequency f_{PLL} . **Table 3-7** shows the output frequency range depending on the K divisor and the VCO frequency range:

Table 3-7 K Divisor Table

K = KDIV+1	K DIV	f_{PLL} for $f_{VCO}=$		Duty Cycle [%]
		400	500	
1	0	400.00	500.00	45-55
2	1	200.00	250.00	50
3	2	133.33	166.67	33
4	3	100.00	125.00	50
5	4	80.0	100.00	40
6...13	5...12	
14	13	28.57	35.71	
15	14	26.67	33.33	46,67
16	15	25.00	31.25	50

Note: Note that the whole range in between two f_{VCO} columns in the above table is allowed.

Note: For divider factors that cause duty cycles far off of 50% not only the cycle time has to be checked, but also the minimum clock pulse width.

When modifying the KDIV value, the PLL logic requires a stable KDIV of maximum 6 cycles of the current output frequency else wise a incorrect frequency is set up (but a glitch will never occur). Possibly occurring metastable conditions caused by an asynchronous start of the switch process are handled properly by the switch logic, but lead to a ± 1 cycle uncertainty in duration of the switch process. To avoid unwanted K Divider settings, KDIV may not be modified while **SYSCON.KDIVRDY** is equal 1.

PLL Lock Detection

The lock detect circuit is part of the PLL. The PLL is a frequency multiplier which generates an output clock signal f_{VCO} , the frequency of which is a multiple of the input frequency f_{IN} . The output signal f_{VCO} is generated by a voltage controlled oscillator VCO, which is regulated to the required frequency via a feedback loop. The feedback loop essentially consists of a divider, which divides f_{VCO} by the PLL (N Factor Divide) factor (the resulting signal is f_{DIV}) and a phase detector, which compares f_{IN} with f_{DIV} . When f_{DIV} has a higher frequency than the reference f_{IN} , then the VCO is steered towards lower

System Control Unit (SCU)

frequency. When, however, the frequency of f_{DIV} is lower than the frequency of the reference f_{IN} , the VCO is steered towards higher frequency. This is how the regulation works. The PLL is made such that the regulation matches the falling edges of the divided PLL clock f_{DIV} and the user-supplied reference clock f_{IN} . As the aspect ratio of f_{IN} is specified in terms of rather flexible bounds and as the aspect ratio of f_{DIV} depends on the PLL factor, this means that the rising edges are not matched at all.

When the regulation is in a stable state, the PLL is said to be locked, else it is said to be unlocked. The task of the lock detect circuit is to watch the PLL (i.e. the signals f_{OSC} and f_{DIV}) and detect a stable lock state. When the PLL is locked, i.e. the regulation has reached a stable state, the lock detect circuit outputs **lock=1**. When the PLL is unlocked (e.g. due to crystal break or just after power on), the lock detect circuit yields **lock=0**.

The lock detect circuit does not interact with the PLL regulation circuitry.

The lock detection supervises the VCO part of the PLL in order to differentiate between stable and instable VCO circuit behavior. The lock detector marks the VCO circuit and therefore the output f_{VCO} of the VCO as instable if the two inputs f_{IN} and f_{DIV} differ too much. Changes in one or both input frequencies below a level are not marked by a loss of lock because the VCO can handle such small changes without any problem for the system. **Table 3-8** shows values below that the lock is not lost for different input values.

Two counters, A and B, count the clock signals of the divided (according to the PLL N division factor) PLL output and the PLL core reference clock f_P . These counters are negative edge triggered and the PLL regulation circuit is constructed such as to match the negative edges of these two clocks.

When, during a counting session, the counters differ by more than two clock periods, the PLL is marked unlocked (**lock = 0**) and the counters are reset to zero. If, however, the counters reach their final value, currently set to $E0_H$, (224) counts and no unlock condition has occurred, the PLL is marked locked (**lock = 1**) and the two counters A and B are again reset to zero.

During sleep mode and after leaving sleep mode, **lock = 0** until the PLL has settled. During power up **lock = X** for a maximum of 6 cycles of either f_P or f_N .

The lock detect circuit is made symmetrical with respect to the permutation of the two input clock signals. It therefore consists of two independent lock detect systems A and B, which are separately controlled by the two counter A and B, respectively. Each system A and B has a separate lock detect output, *lock_A* and *lock_B*, respectively. The resulting lock detect signal **lock** is active only when both *lock_A* and *lock_B* are active. During reset, **lock** is forced active.

The lock detect circuit has the following features:

1. The phase margin (i.e. the tolerance band for **lock=1** result) is independent of the rising edges of the input clocks.
2. The phase margin is permanently watched. Any violation immediately yields **lock=0**. No compensation is possible.

System Control Unit (SCU)

- There are no exceptional time points. **lock=1** only notifies that during the past 192 counts no phase margin violation has been detected.

In order to obtain reliable lock detect information from this circuit after power down, the final counter value (currently $E0_H$), which determines the time T_{cs} of a complete counter session (one count from 0 to 224), is properly adjusted to the regulation behavior of the PLL. During regulation, the VCO frequency first reaches its nominal value, then it overshoots and finally will perform damped oscillations around the nominal frequency. The time T_{cs} of a counter session should not be significantly smaller than the period of the damped oscillations.

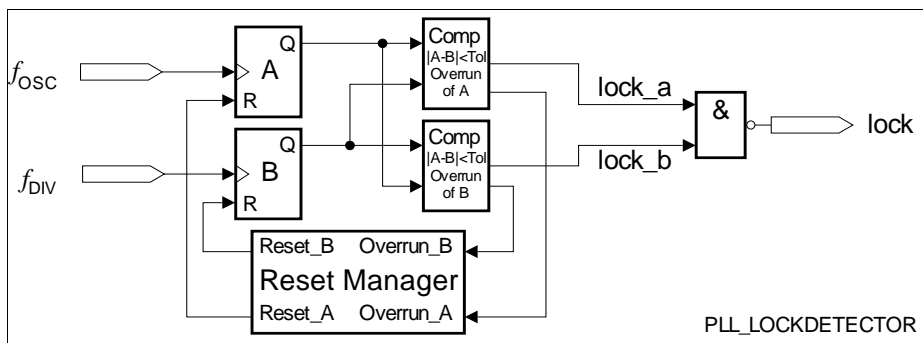


Figure 3-9 PLL Lock Detection

The lock detector marks the VCO circuit and therefore the output f_{VCO} of the VCO as instable if the two inputs f_{REF} and f_{DIV} differ too much. Changes in one or both input frequencies below a level are not marked by a loss of lock because the VCO can handle such small changes without any problem for the system. [Chapter 3-8](#) shows values below that the lock is not lost for different input values.

Table 3-8 Loss of Lock Definition

$\frac{df_{DIV}}{dt}$ for $f_{OSC} =$	
20 MHz	40 MHz
$\leq 14.9 \text{ kHz}/\mu\text{s}$	$\leq 59.5 \text{ kHz}/\mu\text{s}$

Restrictions in changing PLL Configuration Values

The following values/signals are synchronized internally to their corresponding clocks. This avoids unexpected glitches on the output:

System Control Unit (SCU)

- **SYSICON.KDIV** with f_{PLL} : The clock period of f_{PLL} changes after maximum 5 cycles with a rising edge of f_{PLL} . The K-Divider Running Stable Bit (**SYSICON.KDIVRDY** = 0) signals an ongoing **KDIV** reload. While K-Divider Running Stable Bit (**SYSICON.KDIVRDY** = 0) no new value may be written to **SYSICON.KDIV** bitfield.
- **SYSICON.PLLCTRL** with f_{VCO}/f_P ($f_P = f_{OSC}/P$): The clock period of f_{PLL} changes after maximum 7 cycles with a rising edge of f_{PLL} .

All the other signals (e.g. **OSCCON.OSCDISC**, **SYSICON.NDIV**, **SYSICON.PLLVB**) are not synchronized, so any change on these signals may cause glitches on the output clock. Therefore these values must only be changed while the PLL is bypassed, e.g. **SYSICON.PLLCTRL** = 01_B.

In general the K divider influence the duty cycle of the clock f_{PLL} . Because the E-Ray module requires a duty cycle of 50%, only odd values of PDIV are allowed if using the E-Ray module. Furthermore in this application the oscillator frequency, f_{osc} , isn't divided by a factor of P. (The P division ratio must be set to 1 and therefore only defined for this single value. PDIV must be set to 0). Thus the phase detector input frequency, f_P , is always between 20 and 40 MHz.

In this product (SAK-CIC310-OSMX2HT) only one VCO range is available. The VCO range select (**SYSICON.PLLVB**) must therefore be set to 00_B. The VCO output frequency must be 400...500 MHz by selecting **SYSICON.PLLVB** = 00_B.

In this product the VCO must always run at 480 MHz. The **Table 3-9** shows the possible N loop division factors and gives the valid input frequency for these factors. No other factors are allowed for this product.

Table 3-9 Feedback Loop Division Factors for the VCO at 480 MHz

N-Divider Value	NDIV	$f_{VCO} = 480 \text{ MHz}$
		$f_{osc} =$
20	19	24 MHz
24	23	20 MHz

In this product (SAK-CIC310-OSMX2HT) f_{PLL} is 80 MHz. Therefore the K-Divider must be set to 6 (**KDIV** = 5), as shown in **Table 3-10**.

Table 3-10 Output N Divider Factors for f_{pll}

K-Divider Value	KDIV	$f_{VCO} = 480 \text{ MHz}$	Duty Cycle
		$f_{PLL} =$	
6	5	80 MHz	50%

System Control Unit (SCU)

During normal operation only one timing restriction exist (all other control signals may be changed randomly and at any time). This restriction is not applicable for the application the SAK-CIC310-OSMX2HT targets:

KDIV: The maximum switch rate of **KDIV** signals/value is 6 cycles of the minimum output frequency (15 MHz). Thus the worst case switch rate is min. 400nsec (may be extended). Otherwise it may occur that an erroneous frequency is selected (but a glitch will never occur). The K-Divider Running Stable Bit (**SYSCON.KDIVRDY**) = 0 signals an ongoing **KDIV** reload. While K-Divider Running Stable Bit (**SYSCON.KDIVRDY**) = 0 no new value may be written to **SYSCON.KDIV** bitfield.

Note: Possibly occurring metastable conditions caused by an asynchronous start of the switch process are handled properly by the switch logic, but lead to a ± 1 cycle uncertainty in duration of the switch process.

System and PLL Configuration and Status Register

The System and PLL Configuration and Status Register, holds the hardware configuration bits of the PLL, and provide the control for the N, P and K-Dividers as well as the PLL status information.

The clock generation path is selected via the PLL control register SYSCON.

SYSCON

System and PLL Control Register

(20_H)

Reset Value: 0817 020F_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				ENE RAY	KDIV RDY	DIS ME	SYSCLK		0	NDIV					
r				rw	r	rw	rw		rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SW RST	RES LD	LO CK		PLL BY	PLLCTRL		PLLVB		PDIV			KDIV			
rw	rwh	rh	r	rw	rw		rw		rw			rw			

Field	Bits	Type	Description
KDIV¹⁾	[3:0]	rw	PLL Output Divider (KDIV) Scales the PLL output frequency to the desired CPU frequency. $f_{PLL} = f_{VCO} / (KDIV+1)$ [default: $f_{PLL} = f_{VCO}/8$] Valid values: 1111 _B ...0000 _B (Default: 1111 _B)
PDIV	[6:4]	rw	PLL Input Divider Adjusts the oscillator frequency to the defined input frequency range of the PLL $f_{IN} = f_{OSC} / (PDIV+1)$ [default: $f_{IN} = f_{OSC}/2$] Valid values: 111 _B ...000 _B (Default: 000 _B)
PLLVB	[8:7]	rw	PLL VCO Band Select VCO output frequency Base frequency 00 _B 400...500 MHz ~150...320 MHz (Default) 01 _B reserved 10 _B reserved 11 _B reserved
PLLCTRL	[10:9]	rw ²⁾	PLL Operation Control 00 _B Bypass Voltage Controlled Oscillator (PLL clock multiplier), Voltage Controlled Oscillator is off; Prescaler Mode 01 _B Bypass Voltage Controlled Oscillator (PLL clock multiplier), Voltage Controlled Oscillator is running; Prescaler Mode (Default) 10 _B VCO clock used, input clock switched off; Freerunning Mode 11 _B VCO clock used, input clock connected; Normal Mode
PLLBY	11	rw	PLL Bypass Control 0 PLL operates as defined by bit field CTRL (Default) 1 PLL operates in Bypass Mode <i>Note: Switching this setting may not be glitch free and therefore should be avoided.</i>

System Control Unit (SCU)

Field	Bits	Type	Description
LOCK	13	rh	PLL Lock Status Flag 0 PLL is not locked 1 PLL is locked
RESLD	14	rwh	Restart Lock Detection Setting this bit will reset the PLL lock status flag and restart the lock detection. This bit will be automatically reset to 0 and thus always be read back as 0. 0 No effect. 1 Reset lock flag and restart lock detection.
SWRST	15	rw	Software Reset Trigger Setting this bit will automatically request and generate a reset. With the reset execution this bit is automatically cleared.
NDIV³⁾	[20:16]	rw	PLL Multiplication Factor (NDIV) ...by which the PLL multiplies its input frequency $f_{VCO} = f_{IN} * (NDIV+1)$ [default: $f_{VCO} = f_{IN} * 8$] Valid values: 11111 _B ...00000 _B (Default: 10111 _B)
SYSCLK	[24:23]	rw	System Clock Selection 00 _B $f_{SYS} = f_{CGU}$ (Default) 01 _B $f_{SYS} = f_{CGU} / 2$ 10 _B $f_{SYS} = f_{CGU} / 4$ 11 _B $f_{SYS} = f_{sample}$
DISME	25	rw	Disable SSC Move Engine (only relevant in Mode 11_B: SSC Host Interface activated) 0 _B SSC Move Engine Activated (Default) 1 _B SSC Move Engine Deactivated
KDIVRDY	26	r	K-Divider Running Stable 0 KDIV value is currently copied to the K divider and therefore the value of KDIV may not be altered (avoid programming of KDIV at this point of time). 1 K divider is stable and may be programmed by setting a new KDIV value. <i>Note: If reprogramming KDIV with a new value while KDIVRDY is equal 0 may result in an incorrect programming of the K-divider.</i>

System Control Unit (SCU)

Field	Bits	Type	Description
ENERAY	27	rw	Enable E-Ray This bit activates or deactivates the E-Ray module. 0 E-Ray Module is disabled 1 E-Ray Module is enabled (Default)
0	12	r	Reserved; undefined read; should be written with 0.
0	[22:21]	rw	Reserved; reading these bits will return the value last written; read as 0 after reset.
0	[31:28]	r	Reserved; read as 0; should be written with 0.

- 1) KDIV value may not be modified while KDIVRDY is equal 0, else wise an undefined programming of the K-divider will result.
- 2) During JTAG Enabled Mode these bit have the type rh instead rw.
- 3) This bitfield is not synchronized within the PLL, so any change on this bitfield/signal may cause glitches on the PLL clock output, unless N-divider is bypassed.

The following values/signals are synchronized internally to their corresponding clocks. This avoids unexpected glitches on the output:

- **SYSCON.KDIV** with f_{PLL} : The clock period of f_{PLL} changes after maximum 5 cycles with a rising edge of f_{PLL} .
- **SYSCON.PLLCTRL** with f_{VCO}/f_P ($f_P = f_{OSC}/P$): The clock period of f_{PLL} changes after maximum 7 cycles with a rising edge of f_{PLL} .

All the other signals (e.g. **OSCCON.OSCDISC**, **SYSCON.NDIV**, **SYSCON.PLLVB**) are not synchronized, so any change on these signals may cause glitches on the output clock. Therefore these values must only be changed while the PLL is bypassed, e.g. **SYSCON.PLLCTRL** = 01_B.

3.3.3.2 Loss of Oscillator Lock Operation

If PLL loses its lock to the external clock/oscillator, it de-asserts its lock bit **SYSCON.LOCK**. Loss of Oscillator Lock is disabled after reset and has to be activated by setting bit **OSCCON.ORDRES** and the PLL loss of lock detection has to be triggered by setting the restart lock detection bit **SYSCON.RESLD** = 1. The PLL behaves as if the Freerunning Mode is selected (even so the **OSCCON.OSCDISC** remains cleared). Due to the missing input clock the phase detector pulls the VCO down to the lower limit frequency, i.e. the free-running frequency f_{free} .

If the PLL is not the system clock source (not operating in Normal Mode) then the loss of lock flag may be used to check for a stable PLL clock. This allows to switch the PLL parameters dynamically.

System Control Unit (SCU)

If the PLL is selected as system clock source (operating in Normal Mode) a loss of lock event is generated and forwarded to the respective Interrupt Selection 0 to Interrupt Selection 3. In addition, the **SYSCON.LOCK** flag is reset. PLL VCO gradually slow down or speeds up. The PLL may slow down to the minimum frequency being the base frequency or to the upper frequency band limit. Emergency routines can be executed with the SAK-CIC310-OSMX2HT clocked with this base frequency, but may fail in case the PLL VCO speeds up generating a system clock being beyond the chip limits.

If the PLL was the clock source the SAK-CIC310-OSMX2HT remains in this loss-of-lock state until the next power-on reset or after a successful lock recovery has been performed.

3.3.3.3 Loss of Oscillator Lock Recovery

If the PLL is in lost the oscillator lock condition, user software can try to re-lock the PLL again by executing the following sequence:

1. Restarting the oscillator run detection by setting bit **OSCCON.ORDRES**
2. Waiting until **OSCCON.OSCR** is set
3. If bit **OSCCON.OSCR** is set, then
 - The Prescaler Mode has to be selected (**SYSCON.PLLCTRL** = 01_B)
 - The oscillator has to be connected to the PLL if disconnected (**OSCCON.OSCDISC** = 0)
 - Setting the restart lock detection bit **SYSCON.RESLD** = 1
 - Waiting until the PLL becomes locked (**SYSCON.LOCK** = 1)
 - When the **SYSCON.LOCK** is set again, the Prescaler Mode can be deselected (**SYSCON.PLLCTRL** = 11_B) and normal PLL operation is resumed.

3.3.4 Clock Multiplex Unit

The Clock Multiplex Unit (CMU) receives the clock selected by the PLL (f_{PLL}). It generates three different clocks out of f_{PLL} : one for E-Ray signal sampling (f_{SAMPLE}), one as system clock (f_{SYS}) and one for E-Ray system clock (f_{SYS_ERAY}).

To enable power reduction, the system clocks (f_{SYS_ERAY}, f_{SYS}) may be scaled down by a factor of 2 ($f_{SYS}=f_{CGU}/2$) or 4 ($f_{SYS}=f_{CGU}/4$). The CMU includes a down converter and a respective multiplexer.

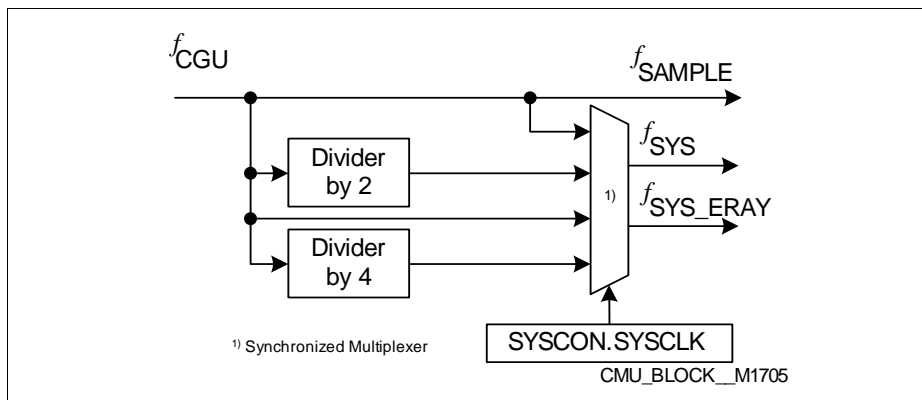


Figure 3-10 CMU Block Diagram

Note: The multiplexer of the CMU Block (SYSCON.SYSCLK) may be switched without clock glitches. Due to this synchronization the clock switching will require at least three f_{SYS} clock cycles to complete.

3.4 Power Supply System

The power supply system is selected that it offers a maximum of flexibility and requires a minimum of pins.

Features

- 3.3V supply for the I/O pads

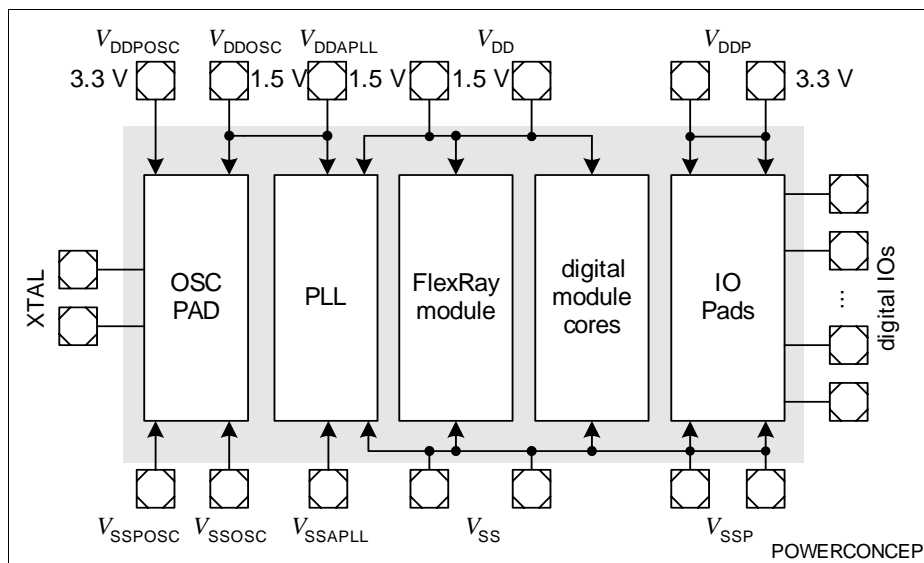


Figure 3-11 SAK-CIC310-OSMX2HT Power Supply System

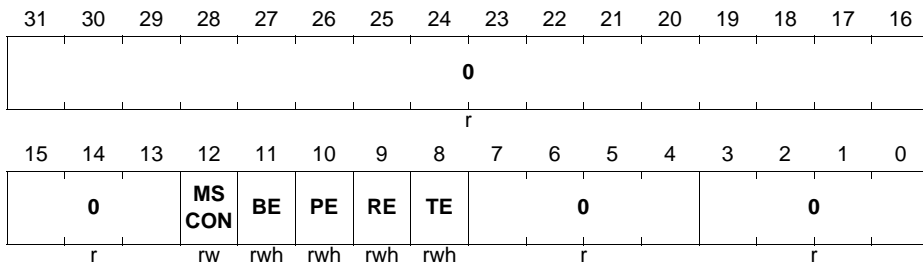
3.4.1 Miscellaneous SCU Registers

SSC_ERRCUM

Cumulative Error Register

(5C_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
TE	8	rwh	Transmit Error Flag 0 _B No error 1 _B Transfer starts with the slave's transmit buffer not being updated Hardware only sets this error flag and software has to reset flag respectively.
RE	9	rwh	Receive Error Flag 0 _B No error 1 _B Reception completed before the receive buffer was read Hardware only sets this error flag and software has to reset flag respectively.
PE	10	rwh	Phase Error Flag 0 _B No error 1 _B Received data changes around the sampling clock edge Hardware only sets this error flag and software has to reset flag respectively.

System Control Unit (SCU)

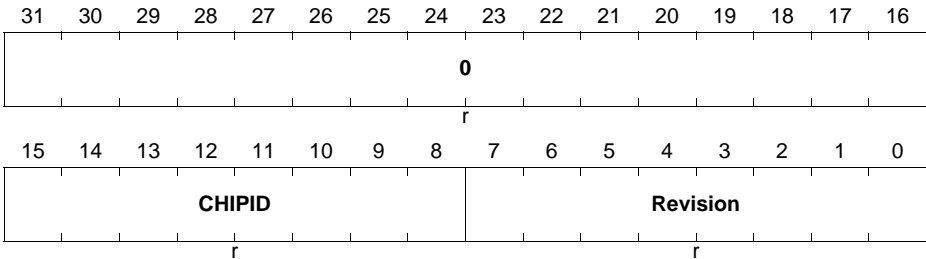
Field	Bits	Type	Description
BE	11	rwh	Baud Rate Error Flag 0_B No error 1_B More than factor 2 or 0.5 between slave's actual and expected baud rate Hardware only sets this error flag and software has to reset flag respectively.
MSCON	12	rw	Master Slave Control 0_B No Half-Duplex Support for SSC. P1_IOC4.PC6 controls the characteristics of P1.6 1_B Half Duplex Support for SSC. Input of P1.5 selects if P1_IOC4.PC6 (P1.5=0) or P1_IOC4.PC6B (P1.5=1) controls the characteristics of P1.6 <i>Note: This bit is only active, if MODE=11_B (SSC Host Interface active).</i>
0	[7:0], [31:13]	r	Reserved; returns 0 if read; should be written with 0.

IDCHIP

Chip Identification Register

(60_H)

Reset Value: 0000 21xx_H



Field	Bits	Type	Description
Revision	[7:0]	r	Device Revision Code Identifies the device step. The revision code is hardwired: 01 _H : SAK-CIC310-OSMX2HT-S AA 02 _H : SAK-CIC310-OSMX2HT-I AA 03 _H : SAK-CIC310-OSMX2HT AB 04 _H : SAK-CIC310-OSMX2HT BA
CHIPID	[15:8]	r	Device Identification Identifies the device name (reference via table). The reset values identify the respective silicon. 21 _H : IFLEX
0	[31:16]	r	Reserved ; returns 0 if read; should be written with 0.

3.5 Interrupt Control

As there is no dedicate interrupt control unit for the IFLEX this task is handled by a small and simply logic inside the SCU. Interrupt are generated towards the system by the XMU, SSC, MLI, DMA, and pins.

3.5.1 Internal Interconnection Signals

In order to simplify the signal naming of the different module outputs or trigger sources, the following names are introduced:

- `eray_int0_o` is the interrupt output 0 of the FlexRay Communication Controller module.
- `eray_int1_o` is the interrupt output 1 of the FlexRay Communication Controller module.
- `eray_tint0_o` is the timer interrupt output 0 of the FlexRay Communication Controller module.
- `eray_tint1_o` is the timer interrupt output 1 of the FlexRay Communication Controller module.
- `mli0_o` is the interrupt output 0 of the MLI module.
- `mli1_o` is the interrupt output 1 of the MLI module.
- `mli2_o` is the interrupt output 2 of the MLI module.
- `mli3_o` is the interrupt output 3 of the MLI module.
- `dma_chn_o` is the channel-related output of the DMA module for channel *n*. This line is activated if `DMA_CHSR0n.TCOUNT` matches `DMA_CHICRn.IRDV`.
- `trig0` is a trigger signal generated from external input signals (with synchronization and edge detection) controlled by bit field `ETCTR.INSEL0`.
- `trig1` is a trigger signal generated from external input signals (with synchronization and edge detection) controlled by bit field `ETCTR.INSEL1`.

3.5.2 External Trigger Inputs

The device supports external trigger sources to start DMA transfers. In general, two internal trigger signals (`trig0`, `trig1`) are available. The device have several input pins capable to deliver a DMA trigger input signal. An edge detection activates the trigger signal upon a event that is configurable via `ETCTR.FENx` and `ETCTR.RENx`.

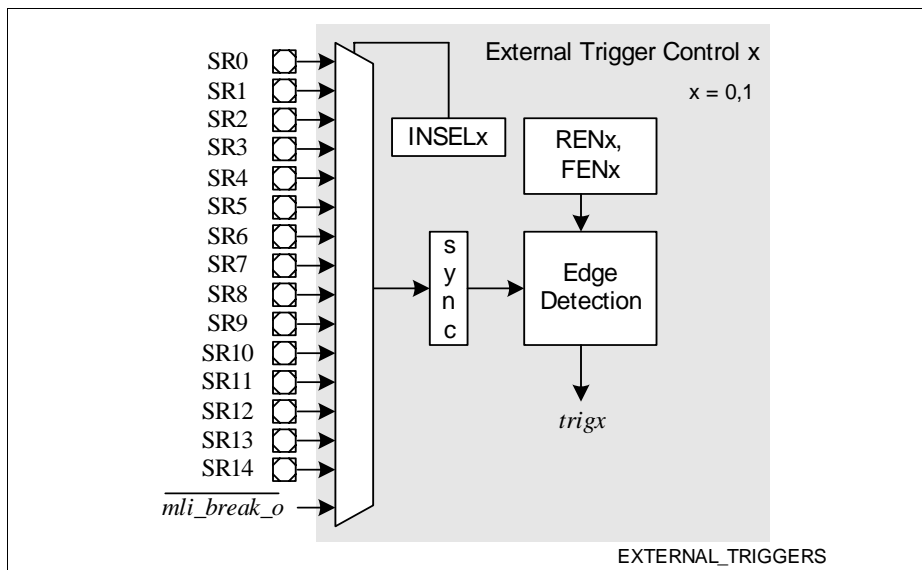


Figure 3-12 External Trigger Unit

The rising edge and falling edge sensitivity of the selected input can be enabled individually. If both edge detections are enabled, a trigger signal is generated upon each change of the signal level (rising edge or falling edge).

Note: Not all possible inputs are necessarily implemented or available in different packages. Please refer to the pin description (see [Table 1-3](#) on [Page 1-21](#)) for details.

The external trigger control register ETCTR contains the bits defining the behavior of the trigger signals TRIG0 and TRIG1.

ETCTR

External Trigger Control Register

(50_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REN 1	FEN 1	0		INSEL1				REN 0	FEN 0	0		INSEL0			
rw	rw	r		rw				rw	rw	r		rw			

Field	Bits	Type	Description
INSEL0	[3:0]	rw	Input Selection for trig0 This bit field defines the trigger source for the trig0. 0000 _B Input SR0 is selected 0001 _B Input SR1 is selected 0010 _B Input SR2 is selected 0011 _B Input SR3 is selected 0100 _B Input SR4 is selected 0101 _B Input SR5 is selected 0110 _B Input SR6 is selected 0111 _B Input SR7 is selected 1000 _B Input SR8 is selected 1001 _B Input SR9 is selected 1010 _B Input SR10 is selected 1011 _B Input SR11 is selected 1100 _B Input SR12 is selected 1101 _B Input SR13 is selected 1110 _B Input SR14 is selected 1111 _B Signal mli_break_o is selected
FEN0	6	rw	Falling Edge Enable for trig0 This bit enables/disables the activation of trig0 upon a falling edge at the selected input. 0 _B The trigger upon a falling edge is disabled. 1 _B The trigger upon a falling edge is enabled.

System Control Unit (SCU)

Field	Bits	Type	Description
REN0	7	rw	Rising Edge Enable for trig0 This bit enables/disables the activation of trig0 upon a rising edge at the selected input. 0 _B The trigger upon a rising edge is disabled. 1 _B The trigger upon a rising edge is enabled.
INSEL1	[11:8]	rw	Input Selection for trig1 This bit field defines the trigger source for the trig1. 0000 _B Input SR0 is selected 0001 _B Input SR1 is selected 0010 _B Input SR2 is selected 0011 _B Input SR3 is selected 0100 _B Input SR4 is selected 0101 _B Input SR5 is selected 0110 _B Input SR6 is selected 0111 _B Input SR7 is selected 1000 _B Input SR8 is selected 1001 _B Input SR9 is selected 1010 _B Input SR10 is selected 1011 _B Input SR11 is selected 1100 _B Input SR12 is selected 1101 _B Input SR13 is selected 1110 _B Input SR14 is selected 1111 _B Signal mli_break_o is selected
FEN1	14	rw	Falling Edge Enable for trig1 This bit enables/disables the activation of trig1 upon a falling edge at the selected input. 0 _B The trigger upon a falling edge is disabled. 1 _B The trigger upon a falling edge is enabled.
REN1	15	rw	Rising Edge Enable for trig1 This bit enables/disables the activation of trig1 upon a rising edge at the selected input. 0 _B The trigger upon a rising edge is disabled. 1 _B The trigger upon a rising edge is enabled.
0	[5:4], [13:12], [31:16]	r	Reserved; read as 0; should be written with 0.

3.5.3 Interrupt Output Structure

The SAK-CIC310-OSMX2HT allows to output internal status or notification signals on output pins. In order to support different applications and pin usage, internal interrupt signals are generated. These signals are then distributed by the port logic and are made available at the service request pins SRn (n = 0...12).

The following event signals can be selected as source for an output of a SRn pin:

- The Interrupt Line 0 of the FlexRay Protocol Controller (eray_int0).
- The Interrupt Line 1 of the FlexRay Protocol Controller (eray_int1).
- The Timer Interrupt Line 0 of the FlexRay Protocol Controller (eray_tint0).
- The Timer Interrupt Line 1 of the FlexRay Protocol Controller (eray_tint1).
- The Transfer Message RAM to Output Buffer RAM Busy of the FlexRay Protocol Controller (eray_TOBC).
- The Transfer Input Buffer RAM to Message RAM Busy of the FlexRay Protocol Controller (eray_TIBC).
- The PLL Loss Lock of the System and Control Unit (PLL_loss_lock)
- The DMA Service Request Outputs (DMA_SRn; n=[6:0], see also [Chapter 5.2.2.2](#))

3.5.3.1 Communication Relations

In case of a MLI communication, there is the possibility to automatically transfer the latest received FlexRay object to the host while the FlexRay is receiving further objects. Depending on the programmed MLI baud rate and the bandwidth of the FlexRay bus, a data transfer is faster than the receiving of new data. As a result, all received FlexRay data can be directly copied to the host without any additional host request. On the other hand, this mode implies that the MLI connection of the host is strictly limited to one SAK-CIC310-OSMX2HT device only. If more than one SAK-CIC310-OSMX2HT device has to be handled by one MLI module on the host device, the data has to be transferred only on request (under full control of the host device).

In the case of a SSC communication, there is a longer time delay between the occurrence of an event and the reaction of the host CPU. Assuming a baud rate up to 5 MBaud, a read request will reach the IFLEX FlexRay register in about 6 μ s (earliest) after the event. This is not enough bandwidth for medium to highly loaded FlexRay buses.

In order to ensure consistency in event handling and result communication, an enhanced interrupt structure has been introduced. It allows to check if a new event occurred while communication for an old was still in progress. Due to the potential delay between an interrupt event and the host reaction, the interrupt signaling is not based of short pulses (as for other devices), but on more static signals. This kind of signal handling allow a similar behavior even for a higher abstraction level, without the risk of lost interrupt events.

3.5.3.2 Service Request Routing

The service request routing allows the user to combine the different interrupt events and status signals as output for the pins SRx. In general, each of the five interrupt output line `intx_o` is monitoring one selectable internal interconnection signal. The alternative data inputs of the pads for the SRx pins are connected as shown in [Figure 3-13](#).

Service Request Signal Length

Because the DMA_SRx only are active for a single cycle, this signal has to be enlarged to generate a active time of 8 clock cycles. Therefore inside the interrupt selection 4 block a counter INTCNT for the control of the interrupt signal length is implemented.

The first rising edge of a DMA_SR signal is activating an internal interrupt state machine and forces the counter INTCNT to the specified value. An additional rising edge does not reactivate the interrupt state machine if already active.

As soon as the interrupt request is inactive, the interrupt counter decrements on every clock pulse. If $INTCNT > 0$, the internal interrupt signal is held active. When reaching 0, the internal interrupt signal is deasserted only if the interrupt request is also deasserted.

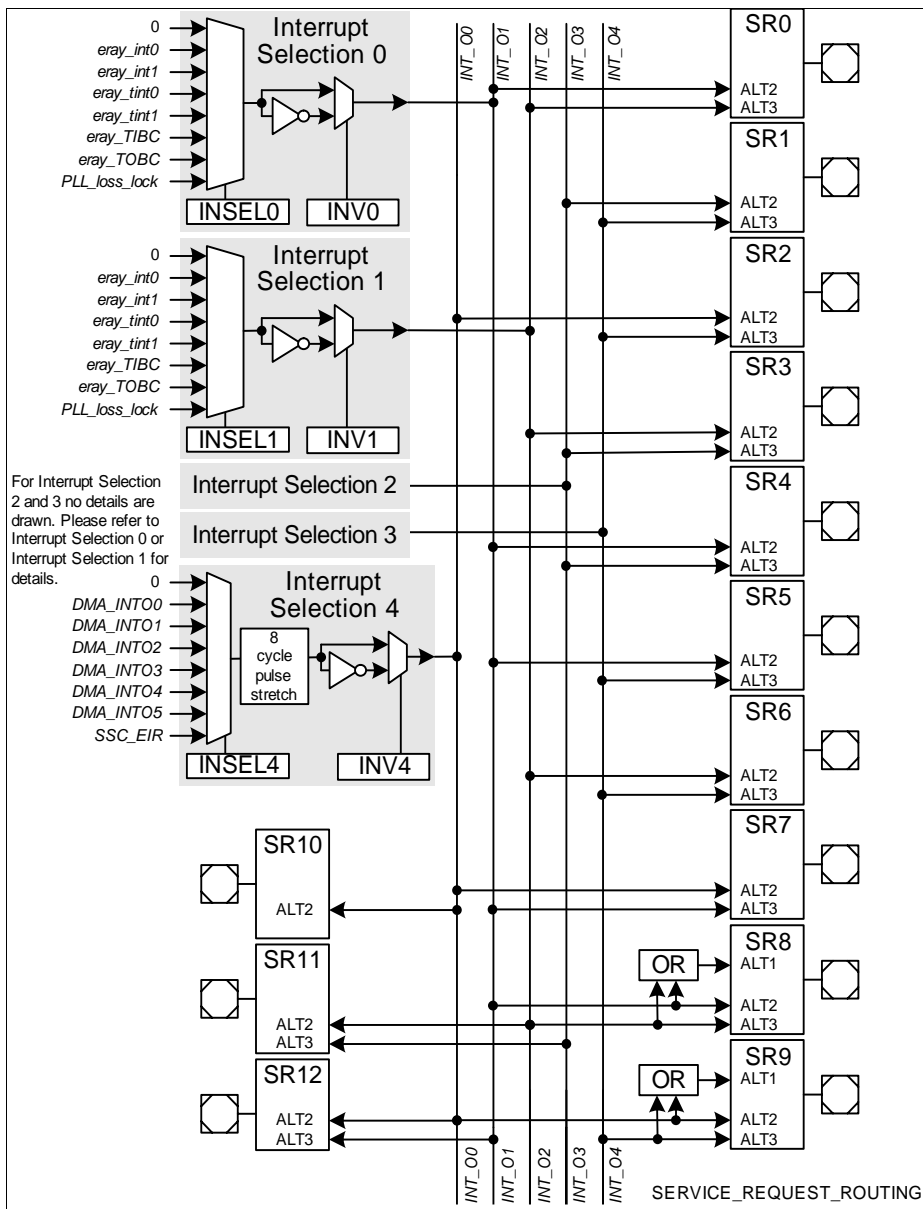


Figure 3-13 Service Request Routing

System Control Unit (SCU)

The SAK-CIC310-OSMX2HT has several service request pins SRx. The internal generated interrupt signals are routed as outputs at these pins. Which status signal is used as source for a interrupt output signal (INTO0 to INTO4) is configurable via the SRCR.INSELx bit fields.

SRCR

Service Request Control Register

(58_H)

Reset Value: 0000 0007_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0												INV4	INSEL4		
r												rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INV3		INSEL3		INV2		INSEL2		INV1		INSEL1		INV0		INSEL0	
rw		rw		rw		rw		rw		rw		rw		rw	

Field	Bits	Type	Description
INSEL0	[2:0]	rw	Input Selection for INTO1 This bit field selects the status signal that is propagated as INTO1. 000 _B The interrupt signal is a permanent 0 001 _B The interrupt signal source is eray_int0 010 _B The interrupt signal source is eray_int1 011 _B The interrupt signal source is eray_tint0 100 _B The interrupt signal source is eray_tint1 101 _B The interrupt signal source is eray_TIBC 110 _B The interrupt signal source is eray_TOBC 111 _B The interrupt signal source is PLL_loss_lock
INV0	3	rw	Invert INTO1 This bit allows an inversion of the signal INTO1. 0 No inversion (active high) 1 Inversion (active low)

System Control Unit (SCU)

Field	Bits	Type	Description
INSEL1	[6:4]	rw	Input Selection for INTO2 This bit field selects the status signal that is propagated as INTO2. 000 _B The interrupt signal is a permanent 0 001 _B The interrupt signal source is eray_int0 010 _B The interrupt signal source is eray_int1 011 _B The interrupt signal source is eray_tint0 100 _B The interrupt signal source is eray_tint1 101 _B The interrupt signal source is eray_TIBC 110 _B The interrupt signal source is eray_TOBC 111 _B The interrupt signal source PLL_loss_lock
INV1	7	rw	Invert INTO2 This bit allows an inversion of the signal INTO2. 0 No inversion (active high) 1 Inversion (active low)
INSEL2	[10:8]	rw	Input Selection for INTO3 This bit field selects the status signal that is propagated as INTO3. 000 _B The interrupt signal is a permanent 0 001 _B The interrupt signal source is eray_int0 010 _B The interrupt signal source is eray_int1 011 _B The interrupt signal source is eray_tint0 100 _B The interrupt signal source is eray_tint1 101 _B The interrupt signal source is eray_TIBC 110 _B The interrupt signal source is eray_TOBC 111 _B The interrupt signal source PLL_loss_lock
INV2	11	rw	Invert INTO3 This bit allows an inversion of the signal INTO3. 0 No inversion (active high) 1 Inversion (active low)

System Control Unit (SCU)

Field	Bits	Type	Description
INSEL3	[14:12]	rw	Input Selection for INTO4 This bit field selects the status signal that is propagated as INTO4. 000 _B The interrupt signal is a permanent 0 001 _B The interrupt signal source is eray_int0 010 _B The interrupt signal source is eray_int1 011 _B The interrupt signal source is eray_tint0 100 _B The interrupt signal source is eray_tint1 101 _B The interrupt signal source is eray_TIBC 110 _B The interrupt signal source is eray_TOBC 111 _B The interrupt signal source PLL_loss_lock
INV3	15	rw	Invert INTO4 This bit allows an inversion of the signal INTO4. 0 No inversion (active high) 1 Inversion (active low)
INSEL4	[18:16]	rw	Input Selection for INTO0 This bit field selects the status signal that is propagated as INTO0. 000 _B The interrupt signal is a permanent 0 001 _B The interrupt signal source is DMA_SR0 010 _B The interrupt signal source is DMA_SR1 011 _B The interrupt signal source is DMA_SR2 100 _B The interrupt signal source is DMA_SR3 101 _B The interrupt signal source is DMA_SR4 110 _B The interrupt signal source is DMA_SR5 111 _B The interrupt signal source is ssc_EIR
INV4	19	rw	Invert INTO0 This bit allows an inversion of the signal INTO0. 0 No inversion (active high) 1 Inversion (active low)
0	[31:20]	r	Reserved ; read as 0; should be written with 0.

Note: The ERAY_TIBC and ERAY_TOBC may generate pulses with a very short duration. Therefore slow pads on the Host or SAK-CIC310-OSMX2HT may lead to a signal distortion making a usage of this signal impossible.

3.5.4 DMA Channel Trigger

All DMA channels n (with n = 0 to 7) are built in the same way (for DMA details, see [Chapter 5](#)). Each DMA channel has an input multiplexer to select one input trigger

System Control Unit (SCU)

source for the activation of the programmed move operation(s). The channels have different input triggers to enable a highly flexible trigger configuration. These differences are sketched in **Figure 3-14** as dotted lines.

In addition, each DMA channel has its own trigger block. This block contains an input trigger selection and two status flags (CHTRn.TF, CHTRn.RF). The Trigger Flag TF is used to store an incoming trigger request, e.g. a DMA channel requesting the move of a data block element, a FlexRay error or status event, or an external trigger activation. TF is cleared automatically by the DMA after the requested transfer(s) is finished. The TF flag is set by signal stf, corresponding to the trigger input selected by bit field CHTRn.TRSEL.

The Ready Flag RF indicates that the MLI is ready to take a new data word to be transferred. An AND-combination of both flags allows to start a DMA move operation if new data is available and the MLI is ready for a new transfer. Flag RF is cleared automatically when the DMA channel operation is triggered.

The DMA channel can also be triggered directly without using both flags. The input multiplexer in each DMA channel can be programmed to select the desired mode.

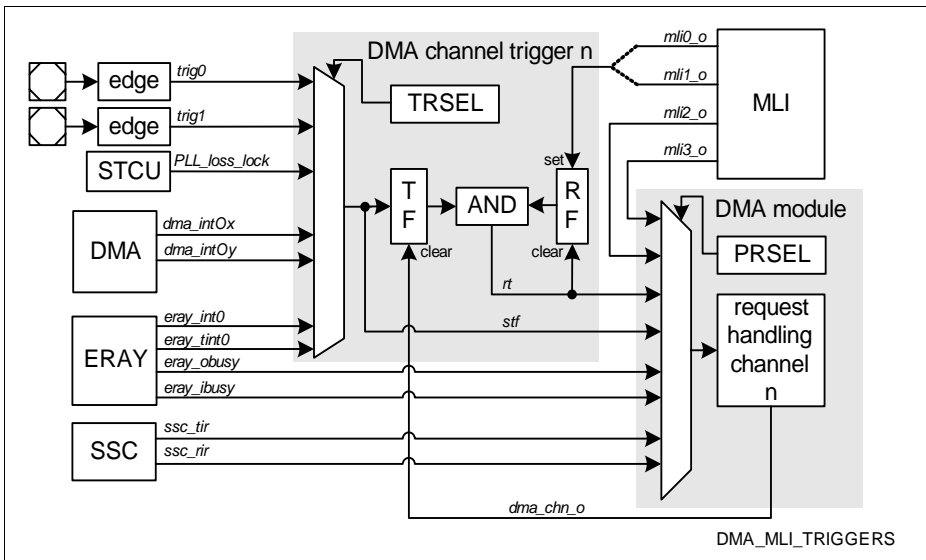


Figure 3-14 DMA Trigger Logic

Note: RF is set by signal mli0_o for the DMA channels 00, 02, 04, and 06. It is set by signal mli1_o for the DMA channels 01, 03, 05, and 07. If the SSC or XMU is selected as communication channel, RF is never set.

System Control Unit (SCU)

The DMA trigger block for each channel is controlled by a channel trigger register CHTRn for channel n (with n = 0 to 7).

CHTRn (n = 0-7)

Channel Trigger Register n (n = 0-7) (30_H + n*4)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								TF	RF	0			TRSEL		
r								rh	rh	r			rw		

Field	Bits	Type	Description
TRSEL	[2:0]	rw	Trigger Selection This bit field defines the trigger source for the DMA channel n. 000 _B A constant 0 is selected. TF and RF are cleared. 001 _B Signal trig0 is selected as trigger source 010 _B Signal trig1 is selected as trigger source 011 _B Signal eray_int0 is selected as trigger source 100 _B Signal eray_tint0 is selected as trigger source 101 _B Signal PLL_loss_lock is selected as trigger source source 110 _B Signal dma_intOn is selected as trigger source 111 _B Signal dma_intOx is selected as trigger source (x=n+ 8)
RF	6	rh	Ready Flag This bit indicates if the MLI is ready for the next transfer. 0 The MLI is not yet ready for a new transfer (former transfer not yet finished) 1 The MLI is ready for a new transfer (former transfer is finished)

Field	Bits	Type	Description
TF	7	rh	Trigger Flag This bit indicates that a channel trigger request is pending. 0 No channel trigger request is pending 1 A channel trigger request is pending
0	[5:3], [31:8]	r	Reserved ; read as 0; should be written with 0.

3.6 Mode Selection

There are two dedicated mode select pins for the SAK-CIC310-OSMX2HT that select the operating mode during the $\overline{\text{PORST}}$ reset sequence.

3.6.1 MODE 0 and Mode 1 Pin

The MODE 0 and Mode 1 pins define if the MLI, XMU, or SSC communication interface is activate after RESET.

The following modes are supported:

- $\text{MODE} = 00_{\text{B}}$: On-chip XMU interface is activated and configured as only communication interface.
- $\text{MODE} = 01_{\text{B}}$: On-chip MLI: On-chip MLI interface is activated and configured as only communication interface.
- $\text{MODE} = 10_{\text{B}}$: On-chip XMU interface is activated and configured as only communication interface.
- $\text{MODE} = 11_{\text{B}}$: On-chip SSC: On-chip SSC interface is activated and configured as only communication interface.

Mode 0 is the LSB of Mode and Mode 1 the MSB. Mode 1=0 and Mode 0 = 1 equals Mode = 01.

3.6.2 $\overline{\text{JTAGEN}}$ Pin

This pin selects between the Normal Mode and a special JTAG Enabled Mode.

The pin $\overline{\text{JTAGEN}}$ defines if the device should enter a JTAG Enabled Mode ($\overline{\text{JTAGEN}} = 0$) or should operate in Normal Mode ($\overline{\text{JTAGEN}} = 1$).

The JTAG Enable Mode is not to be used by a customer. It is only used for production test and field failure analysis by Infineon.

3.6.3 XMU Global Control Register

The XMU Global Control Register XMU_CON provides global control of the XMU. The XMU Global Control Register XMU_CON allows to enable/disable the XMU in general. After reset the XMU is enabled.

XMU_CON

XMU Global Control Register

(64_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			ADREXT												ADR C
r			rw												rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												BCA0S		EXT ACC	0
r												rw		rw	r

Field	Bits	Type	Description
EXTACC	1	rw	External Access to Crossbar Switch Control 0 External accesses to the Crossbar Switch are disabled (default after reset). 1 External accesses to the Crossbar Switch are enabled.
BCA0S	[3:2]	rw	Byte Control and Address 0 Select 00 _B 16 data Lines: XMU BC[1:0] is hardwired to 0 (V _{DD}) and a 16 Bit Data Interface is activated 01 _B 16 data lines: XMU BC[0] is connected to port pin P1.0, XMU BC[1] is connected to port pin P1.7, and a 16 Bit Data Interface is activated 10 _B (reserved) 11 _B 8 data lines: A[0] is hardwired to 0 and a 8 Bit Data Interface is activated (D[15:8] unused)

System Control Unit (SCU)

Field	Bits	Type	Description
ADRC	16	rw	<p>External Address Extension Control</p> <p>0 XMU A[23:13] are hardwired to 0 (V_{SS}). A[11] is connected to port pin P2.11 and A[10] to P2.10.</p> <p>1 XMU A[21:11] are defined by the bitfield XMU_CON.ADREXT[10:1]. A[23] is connected to port pin P2.11 and A[22] to P2.10.</p> <p>In JTAG ENABLE Mode:</p> <p>0 XMU A[23:12] are hardwired to 0 (V_{SS}). A[10] is connected to port pin P2.10 and A[9] to P2.9.</p> <p>1 XMU A[21:10] are defined by the bitfield XMU_CON.ADREXT[10:0]. A[23] is connected to port pin P2.10 and A[22] to P2.9.</p> <p><i>Note: After setting ADRC to 1, XMU_CON may no longer be accessible via XMU. To reset ADRC an internal DMA channels may have to be set up before and triggered by an external event line to reset ADRC respectively.</i></p>
ADREXT	[28:17]	rw	<p>Address Extension</p> <p>The bitfield ADREWXT[11:1] defines the value of the XMU address lines A[21:11] in case XMU_ADRC is set to 1.</p> <p>In JTAG ENABLE Mode This bitfield defines the value of the XMU address lines A[21:10] in case XMU_ADRC is set to 1.</p> <p><i>Note: The LSB of ADREXT (ADREXT[0]) is only used in JTAG ENABLE Mode.</i></p>
0	0, [15:4], [31:29]	r	Reserved ; read as 0; should be written with 0.

Note: A[21:13] are controlled by the XMU_CON register (A[21:12] in JTAG ENABLE Mode)

Note: A[23:22] are either controlled by the external pins or hardwired to 0.

Note: A[12:11] are either controlled by the XMU_CON register or hardwired to 0 (A[11:10] in JTAG ENABLE Mode).

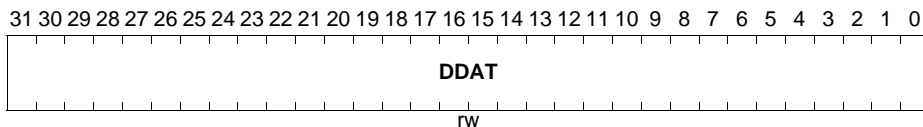
3.6.4 DMA Reprogramming

The eight DMA channels may not be sufficient for an autonomous transfer of receive and transmit of FlexRay data to the host microcontroller via MLI. Therefore a re programming of DMA channels by other DMA channels may be required. The data for the reprogramming may be stored into the registers DMADATnn (with nn = 00 to 31). These register do not configure any hardware function directly, but act like SRAM Cells to store data to be transferred by a DMA channel into registers, e.g. DMA registers.

DMADATnn (nn = 00-31)

DMA Data Register nn (nn = 00-31) ($80_H + nn * 4$)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
DDAT	[31:0]	rw	Data Field for DMA Data Field to hold data to reprogram DMA channels.

3.6.5 Gather Scattered Bits

Message Objects may contain packed bitfields. Because the data width handled by DMA is either 8 Bit, 16 Bit, or 32 Bit, bitfields with different widths cause a communication overhead, especially when building up gateways to CAN communication networks. Therefore the SAK-CIC310-OSMX2HT has a circuitry to gather scattered bitfields into a Byte or Halfword. This circuitry is built up by a 32 bit input registers GSBIR (Gather Scattered Bits Input Register) to store the data containing packed bitfields. A mask stored in the Mask Register GSBMR (Gather Scattered Bits Mask Register) selects respective bits in the GSBIR Register and stores them in the Output Register GSBOR (Gather Scattered Bits Output Register). So all bits in the GSBIR register are transferred into the GSBOR register, if the respective bit in the GSBMR Mask register is equal 1. All transferred bits are shifted right to fill up the gaps of non selected bits.

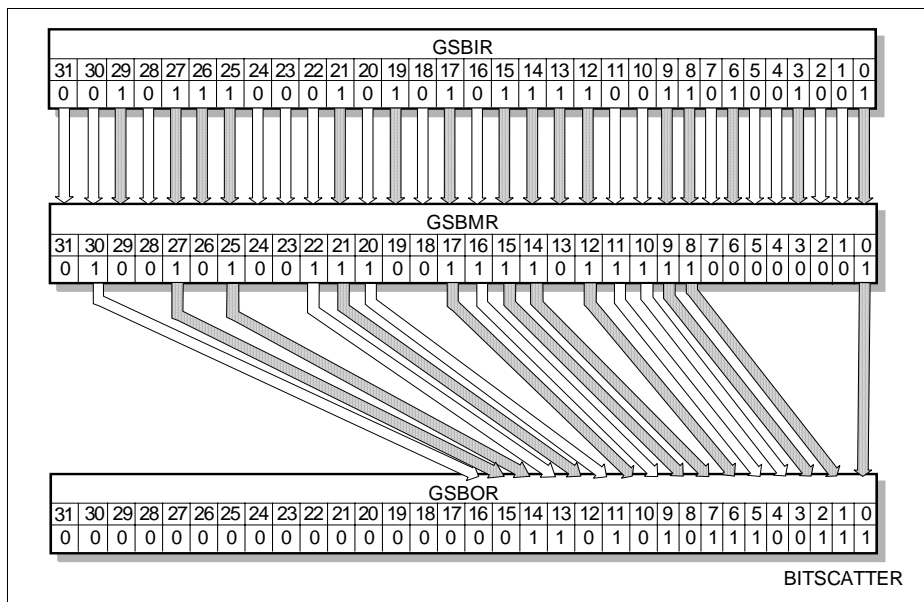


Figure 3-15 Bit Scatter Bit Gather Logic

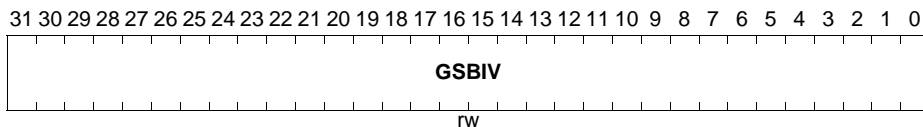
Example:

GSBIR=2E2A F349_H and GSBMR=4A73 DF01_H results in a GSBOR=0000 6AE7_H

GSBIR

Gather Scattered Bit Input Register (68_H)

Reset Value: 0000 0000_H

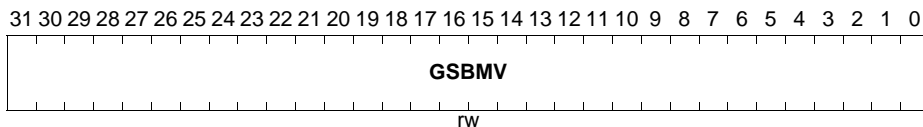


Field	Bits	Type	Description
GSBIV	[31:0]	rw	Gather Scattered Bit Input Value Data Field to enter Packed Data.

GSBMR

Gather Scattered Bit Mask Register (70_H)

Reset Value: 0000 0000_H

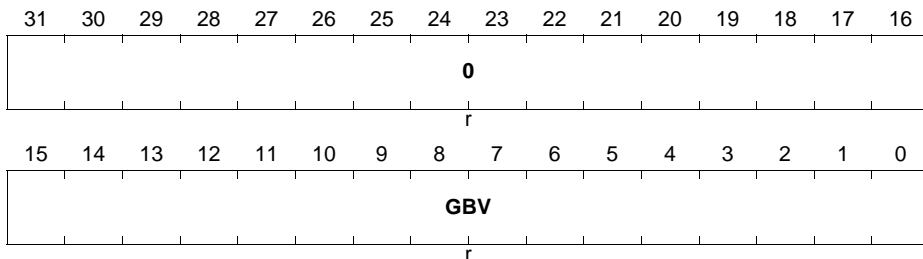


Field	Bits	Type	Description
GSBMV	[31:0]	rw	Gather Scattered Bit Mask Value Data Field to select Packed Data Bits.

GSBOR

Gather Scattered Bit Output Register (78_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
GBV	[15:0]	r	Gathered Bit Value Right Aligned Packed Data Bits.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

3.6.6 Address Map

In the FlexRay Communication Controller, the registers of the STCU module are located in the following address Range:

- Module Base Address = 0000 0800_H
Module End Address = 0000 08FF_H
- Absolute Register Address = Module Base Address + Offset Address (offset addresses see [Table 3-2](#))

4 Host Communication Channels

Three different communication channels are implemented within the SAK-CIC310-OSMX2HT enabling the communication with an external host controller via either one of the two serial interfaces or the parallel communication channel.

In general the IFLEX is a 32-bit orientated architecture performing 8, 16, and 32 bit operations. Four modules are responsible (masters) for all data traffic operations on the IFLEX:

- **DMA:** The DMA may handle 8,16, and 32 bit read and write accesses.
- **MLI Move Engine:** The MLI Move Engine may operate 8,16, and 32 bit read and write accesses.
- **SSC Move Engine:** The SSC Move Engine may perform 16 bit read and write accesses only.
- **XMU (Parallel External Interface):** The XMU is able to generate 8 bit and 16 bit read and write accesses.

4.1 Host Write Request

The host requests one of the host communication interfaces (as selected by the mode pins) to write data to a certain address. The corresponding move engine is activated, reads the data from the receive register, buffers it within the move engine, and writes it to the respective target address.

4.1.1 MLI Write Operation

A CPU write access to a transfer window of the host MLI transfers the data to the remote MLI receiver on the IFLEX. The received information is stored within remote MLI receiver registers. There it is accessible for the MLI move engine bus master to transfer the data to the requested destination address.

The registers of the remote MLI receiver used by a host write operation:

Table 4-1 MLI Write Operation Register

Register	Bit Field	Name
Receiver Address Register	Address	MLI_RADDR.ADDR
Receiver Data Register	Data	MLI_RDATAR.DATA
Receiver Control Register	Data Width	MLI_RCR.DW
Receiver Control Register	Received Pipe Number	MLI_RCR.RPN
Receiver Control Register	Type of Frame	MLI_RCR.TF
Receiver Interrupt Status Register	Normal Frame Received Interrupt Flag	MLI_RISR.NFRI

Host Communication Channels

The signal `rqt` indicates to the MLI move engine the reception of new data in the MLI receive registers. The `MLI_RCR.TF` bit signals to the move engine signals either a read or write operation and the `MLI_RCR.DW` provides the data width. The MLI move engine reads the received data from `RDATAR.DATA` and write the respective number of byte to the address available in `MLI_RADDR.ADDR`. After reading the `MLI_RADRR` and `MLI_RDATAR` registers the MLI interface asserts the signal `gnt`.

4.1.2 SSC Write Operation

Out of the SSC serial data stream the SSC move engine autonomously extracts the control data. It can handle single and block (multiple) halfword transfers. The SAK-CIC310-OSMX2HT only handles serial data length of 16 bit (halfwords).

For the following description see [Figure 4-14](#), [Figure 4-15](#) and [Figure 4-16](#). For each write transfer received by the SSC slave interface of the SAK-CIC310-OSMX2HT, the protocol engine extracts the following control data from the first received halfword: The command `CMD` bit field, the increment bit `INCE`, and the address `ADDR` bit field. The received halfwords following the first command halfword are handled as reception data stream. On availability of `CMD`, `INCE`, and `ADDR` bit field the move engine bus master transfer the reception data stream to the destination address/addresses. Because only having a single halfword buffer the move engine transfers each received data stream halfword (two byte) immediately after its reception.

4.1.3 XMU Write Operation

The host needs to have ownership of the external parallel bus; thus, bus arbitration will be required in many cases ahead of such an access. This master can access the XMU by activating the XMU chip select input `CSFPI` and presenting a proper address on the external address bus. This address is forwarded together with the data on the external data bus to the internal crossbar switch selecting a respective register. This register latches (reads) the data on the internal cross bar switch.

4.2 Host Read Request

The host requests to read data from an address within the SAK-CIC310-OSMX2HT via the activated (mode pin) serial interface. The move engine of this serial interface reads the requested data from the source address and writes it in to the respective transmit register.

4.2.1 MLI Read Operation

A host CPU read access to a transfer window of the host MLI triggers a command frame transfer to the remote MLI receiver of the IFLEX. The received command frame is stored in the MLI receiver data register of the IFLEX (SAK-CIC310-OSMX2HT). The MLI move engine accesses this data to read the requested data from the source address.

Host Communication Channels

The registers of the remote MLI receiver used by a host read request are:

Table 4-2 MLI Read Operation Register

Register	Bit Field	Name
Receiver Address Register	Address	MLI_RADDR.ADDR
Transmitter Status Register	Answer Pipe Number	MLI_TSTATR.APN
Receiver Control Register	Data Width	MLI_RCR.DW
Receiver Control Register	Received Pipe Number	MLI_RCR.RPN
Receiver Control Register	Type of Frame	MLI_RCR.TF
Receiver Interrupt Status Register	Normal Frame Received Interrupt Flag	MLI_RISR.NFRI

The signal `rqt` notifies the MLI move engine of new data in its registers. The `MLI_RCR.TF` bit signals to the move engine signals read or write operation and the `MLI_RCR.DW` provides the data width. The MLI move engine reads the respective number of bytes from the address available in `MLI_RADDR.ADDR` and writes it to the `TDRAR` (Transmitter Data Read Answer Register) register. After reading the `MLI_RADDR` register the MLI interface asserts the signal `gnt`. The action of the move engine is terminated by `MLI_RISR.MEI` (Move Engine Access Terminated Interrupt) and `TRSTATR.AV = 1` (Answer Valid) and the transmitter transfers the requested data to the host MLI receiver.

4.2.2 SSC Read Operation

The SSC communication protocol may be used for single halfword and block (multiple halfword) transfer. For the following description see [Figure 4-11](#), [Figure 4-12](#) and [Figure 4-13](#). For each write transfer received via the SSC interface the protocol engine extracts the relevant control data out of the first received halfword. This is the command `CMD`, the increment indicator `INCE`, and the address `ADDR`. Using the address control data the move engine reads the source data and writes it to the SSC transmit buffer `SSC_TB`. The move engine writes each halfword after reading it, without temporary buffering it in the move engine.

4.2.3 XMU Read Operation

The host needs to have ownership of the external parallel bus; thus, bus arbitration will be required in many cases ahead of such an access. This master can access the XMU by activating the XMU chip select input `CSFPI` and presenting a proper address on the external address bus. This address is forwarded to the internal crossbar switch selecting a respective register. This register presents the requested data on the internal cross bar

Host Communication Channels

switch. The XMU forwards the data from the internal crossbar switch to the external data bus.

4.3 SSC Move Engine

The initial state of the SSC Move Engine (ME) is state A. By the activation of pin $\overline{\text{SLS}}$ the ME changes to state B. When the first halfword has been received state C is entered. The first halfword is the transaction header composed by the CMD bit, the INCE bit, and the 14 bit address ADDR bit field. The header parameter INCE = 1 defines the following data halfwords to be written to successive addresses, INCE = 0 defines the following data all to be written to the same address. CMD = 1 (write transfer) switches the ME to state W1 and waits for valid data available in the SSC receive buffer (SSC_RB). On arrival of valid data RIR is activated and the ME passes to state W2, where the value read from SSC_RB is written to the address ADDR. Conditionally the ADDR is increment in respect to INCE and the ME returns to state W1. This process is continued until $\overline{\text{SLS}}$ is deactivated. If CMD = 0 (read transfer) the ME switches to state R1 and reads the data from the requested address ADDR and write it to the SSC transmit buffer (SSC_TB). After storing the data the R2 state is entered, and TIR is activated after transmitting the data by the SSC. The ADDR is increment according to INCE. This process is continued until $\overline{\text{SLS}}$ is deactivated.

SSC Transaction Header

The following table defines the SSC transaction header for write transfers.

Table 4-3 Transaction Header

Name	Description	Bit Position
CMD	Transfer Type Identifier 0 A read transfer is selected 1 A write transfer is selected	15
INCE	Address Increment Enable 0 The destination address is not increment after each transaction 1 The destination address is increment after each transaction	14
ADDR	Destination Address	13...0

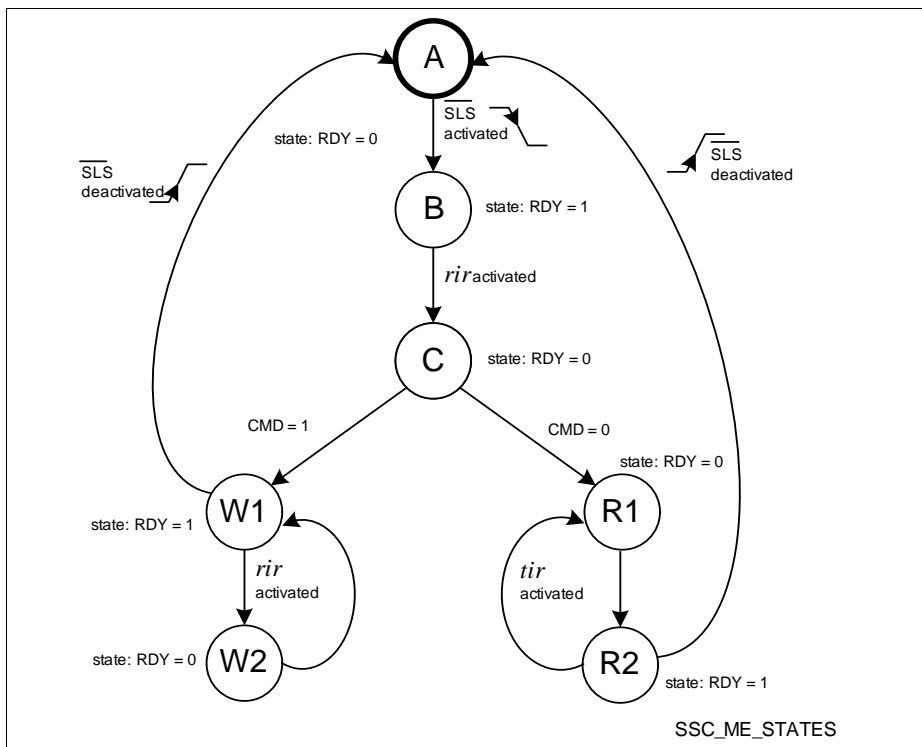


Figure 4-1 States of the SSC Move Engine

To ensure a reliable communication the host master synchronizes to the availability of the requested data through an additional signal. The host master continues and stops the activation/deactivation of the SSC master clock SCLK respectively to this additional synchronization signal. As the ME requires some cycles to move the requested data from the source to the transmit buffer, the master has to stop the SSC clock until the requested data is available in SSC_TB. In case of a write the slave (IFLEX) may require extra time to move the received data from the receive buffer to the destination address. This synchronization is provided by the RDY pin.

A transition of the RDY signal from de-asserted to asserted signals the SSC master to reactivate the SSC clock (SCLK) as the slave is ready continue the communication. A transition of the RDY signal from asserted to de-asserted signals the reception of the first SSC clock cycle (by the SSC slave) to the SSC master.

The RDY pin is controlled by the ME in the following manner:

Host Communication Channels

RDY remains de-asserted if the ME is in state A. When $\overline{\text{SLS}}$ is asserted by the SSC master the ME asserts RDY to acknowledge the SLS assertion (ME transition from state A to state B). The SSC master may now provide the SSC clock SCLK. After receiving the first SCLK cycle the SSC slave transmits the transmit buffer content into the shift register and generates the TIR signal. While de-asserting the TIR signal the RDY signal is also de-asserted. The further flow now differs in case of a read (Read Case) or write (Write Case) request by the SSC Master.

Read Case

In case of a read request the ME copies the requested read data from the requested address to the slave SSC transmit buffer and asserts in parallel the RDY signal again. The master resumes on sending a second header and the slave start in parallel to send the read data requested by the previous header transfer. After receiving the first SCLK cycle the slave SSC transmits the transmit buffer content to the shift register and generates the TIR signal. In parallel with the de-assertion of the TIR signal the RDY signal is also de-asserted. This process is repeated as long as the master keeps this communication alive.

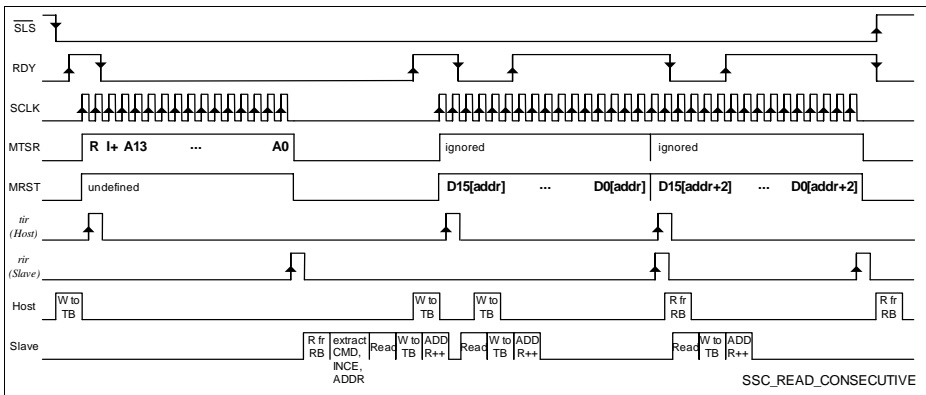


Figure 4-2 Consecutive READ (SSC Move Engine View)

Note: If the SSC of the SSC master (host controller) does not support a RDY signal by hardware, an alternative communication scheme may be used, if limiting the on chip access times to 40 system clock cycles by configuring the ERAY buffer access delay time ($\text{ERAY_CUST3.TO} < 20$) and DMA block size ($\text{DMA_CHCR0n} (n = 0-7). \text{TREL} < 10$) to the respective values. If the delays have been configured respectively the move engine has completed the data read operation within 64 system clock cycles under all circumstances. Therefore the host controller may send after the first 16 bit address word an additional 16 bit dummy value back to back to the first command word transmitted to the slave to insert a time delay large

Host Communication Channels

enough for the move engine to finalize data fetch. After this dummy word additional dummy words may be send back to back by the host to receive the required read data without any further bandwidth loss. This enable the usage of the hardware controlled chip select and makes the evaluation of the RDY signal obsolete, but will in return reduce the data throughput for single data read operations in many cases. **Figure 4-3** sketches this type of read access not evaluating the RDY signal. The master may send back all received data in the respective following communication cycle back to the slave (ignored data) and configure the DMA to compare the data copied from the transmit buffer with the data copied from the receive buffer (received by the slave) a communication cycle later. The chip select for the slave could e.g. be generated by an OR of two different SSC chip select of the master. One of these two chip select is routed as trigger input to the DMA. Therefore the customer may trigger an compare by activating the respective chip select.

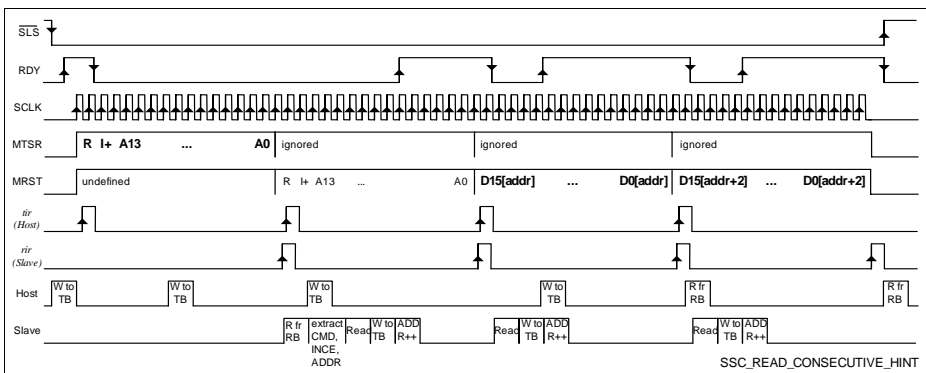


Figure 4-3 Consecutive READ ignoring RDY signal (SSC Move Engine View)

Write Case

In case of a write the ME waits for the RIR signal to be asserted by the SSC slave and then reads the SSC receive buffer. In parallel to the read action of the receive buffer content the ME asserts the RDY signal again. The master may continue the SSC clock SCLK if the RDY signal is reasserted. With the next TIR signal RDY is de-asserted. After the next SSC frame after the assertion of RIR the RB content is read and RDY is asserted. This process is repeated as long as the master keeps the communication alive. To detect communication errors the DMA may be configured to copy the received data into the transmit buffer. The master may then compare the transmitted command word and data words with the data received the respective cycle later. The last transmitted data word is received by the master when sending the next command word to the slave (so while sending the address, Read/Write bit and Increment bit), regardless if this

Host Communication Channels

following transfer is a Read or Write transfer or is even interrupted after transmitting the command word. This enable to detect every single bit communication faults and nearly every double error communication fault.

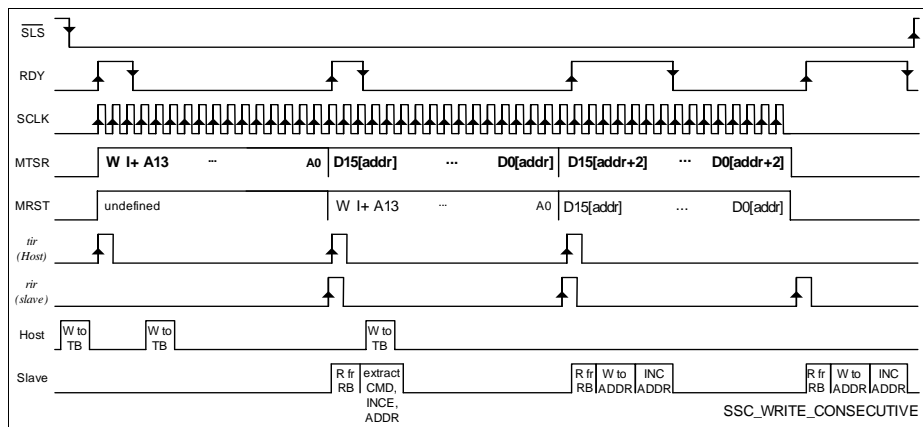


Figure 4-4 Consecutive Write (SSC Move Engine View)

4.4 Communication Principles MLI

The communication principle of the Micro Link Interface allows data to be transferred between a local and a remote controller without intervention of the CPUs. The read mode also allows to ask the other side for the desired data due to a Move Engine which is available on the remote side. It is also possible to interrupt the remote side by sending up to four different commands. **Figure 4-5** illustrates a general overview of two MLIs connected, in which the transfer windows, TW, and the move engine of the remote side have been detailed. As it is possible for the controller on the local side to read and write data to/from an address on the remote side, it is not necessary to define a protocol on the serial connection as it is required for the SSC.

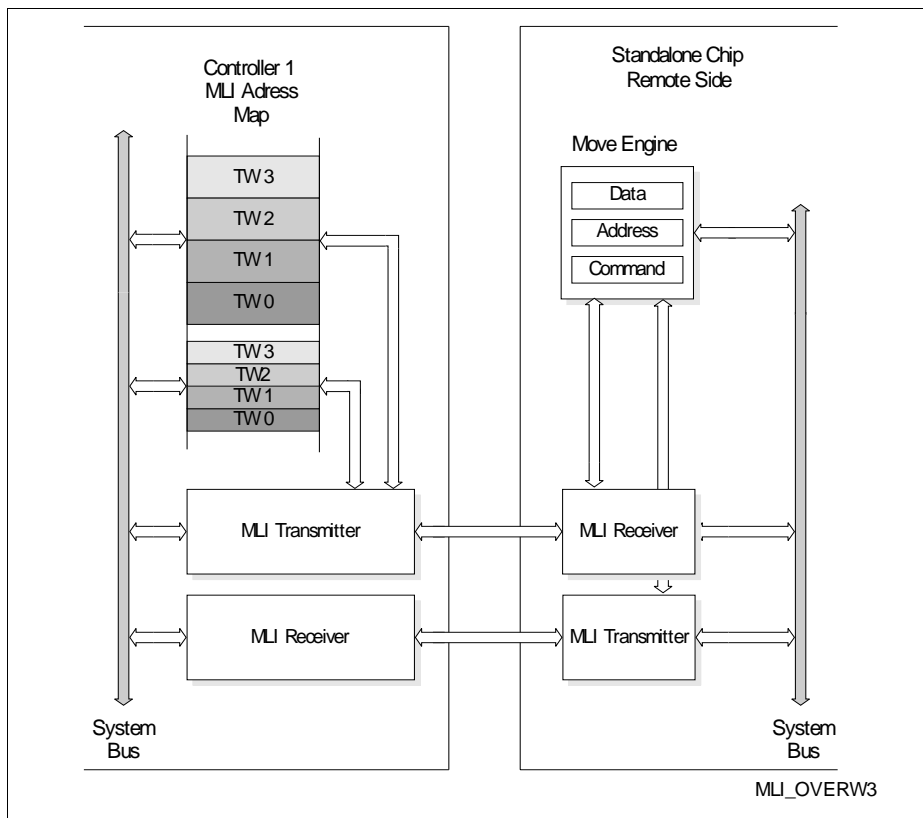


Figure 4-5 MLI Bridge Between a Microcontroller and IFLEX

4.4.1 Prerequisites for Communication

To be able to start the communication it is necessary to enable the receiver on the remote MLI during the PORST procedure. With these prerequisites the controller on the local side is capable to address other registers of the remote MLI.

4.4.2 Enable MLI Interfaces

The local MLI is used as serial communication interface to the SAK-CIC310-OSMX2HT via the remote MLI interface, with a transfer rate of about 15 Mbit/sec during startup, if CPU runs with about 60 MHz. Because the MLI interfaces are cleared with every reset, most initialization steps are performed with every reset.

Host Communication Channels

The local MLI and remote MLI initialization for startup is characterized by the following steps:

- Enable local MLI (clock), adjust baud rate to highest frequency
- Initialize local MLI transfer interface for communication with remote MLI
 - Enable communication from remote side by clearing the receiver reset bit in local MLI
 - Select signal set (interface) #2 for transmit and receive
 - Set parity delay to 6 clocks for transmit and receive timing
- The remote MLI interface works with its default settings. Only the transmitter and receiver have to be enabled via local MLI.
- Install a local MLI/remote MLI pipe for transfers
 - Select a fixed address window (maximum 64K per window) in local MLI for transmit
 - Write related SAK-CIC310-OSMX2HT base address to local MLI: the according remote MLI buffer is initialized automatically by local MLI.

4.4.3 Single steps of Communication via MLI bridge

Transfer addresses from local side to SAK-CIC310-OSMX2HT (and vice versa) are translated on the MLI interface using the following scheme:

- The address from the controller to the local MLI points to the used address window of the local MLI (fixed address)
- The relative SAK-CIC310-OSMX2HT address is transferred as offset to the local MLI.

Data transactions are in principle done by (view of CPU):

- Writing address and data of the target in the remote bus domain to the local MLI.
- Then the transaction is initiated between the 2 MLI's of the bridge.
- The remote MLI, which is master, accesses the remote bus system, reads or writes the data as requested
- In the case of a read transaction the data are then sent back to the local MLI on CPU side. In case of a write transaction the action was "fire and forget".
- After being signaled by relevant flags in the local MLI receiver, the CPU can read the requested data from MLI1_RDATAR. This is initiated either by polling or by interrupt.

4.4.4 DMA Support for MLI (Single Message Buffer Read Support)

Only if using the MLI as Host Interface, the DMA of the SAK-CIC310-OSMX2HT may support the data transfer. This section describes an possible DMA support. [Table 4-4](#) summarizes the initialization values of the respective DMA, MLI, and SCU registers.

4.4.4.1 Single Message Buffer Read Support

This chapter describes the usage of the DMA to read a single user defined message buffer.

Host Communication Channels

First the Host controller has to configure the DMA, SCU, and MLI as summarized in [Table 4-4](#). The last initialization step should configure the MLI_GINTR register to set the respective RF flip-flops. After this initialization, if the Host controller wants to read a respective message buffer, these are the steps the host controller has to do:

1. Host controller has to clear respective (internal) memory location of the semaphore.
2. Host controller has to write via MLI Pipe 0 the requested Message Buffer Number (ERAY_OBCR.OBRS) and set the Request Bit (ERAY_OBCR.REQ) into the Output Command Buffer (ERAY_OBCR).
3. Frequently read the semaphore from respective (internal) memory location and check if transfer is ready. A Interrupt by a pin could be used alternatively.
4. Transfer is completed and data can be read from respective memory location.

The DMA receives a ready signal from the ERAY_OBCR as soon as the requested Message Buffer has been copied into the Output Buffer Shadow.

1. DMA Channel 00 is triggered by this ERAY_OBUSY signal and sets the ERAY_OBCR.VIEW bit by moving a 01_H from SCU_DMADAT01.[7:0] to ERAY_OBCR.[15:8].
2. After DMA Channel 00 completed transfer, DMA Channel 01 is triggered. DMA Channel 01 moves the received payload length of the selected message buffer into the transfer count reload bit field of DMA channel 02 (DMA_CHCR03.TREL).
3. After DMA Channel 01 completed DMA Channel 02 is triggered as soon as the MLI Pipe 0 is ready for a transfer. DMA Channel 02 transfers four 32 Bit data from the ERAY_RDHS0, ERAY_RDHS1, ERAY_RDHS3, and ERAY_MBS registers to the MLI Pipe 0 as write transfer. After every 32 bit transfer, the DMA_Channel waits for the MLI interrupt signalling a completed previous transfer.
4. After DMA Channel 02 completed move, DMA Channel 03 is triggered as soon as the MLI Pipe 1 is ready for a transfer. DMA Channel 03 transfers several 16 Bit data from the RDDS registers as specified in the RDHS2.PLR bit field to the MLI Pipe 1 as write transfer. After every 16 bit transfer, the DMA_Channel waits for the MLI interrupt signalling a completed previous transfer.
5. After DMA Channel 03 completed the four data moves, DMA Channel 05 is triggered. DMA Channel 05 moves two 16bit data from DMADAT00[15:0] and DMDAT00[31:16] to the DMA_SADR02 and DMA_DADR02 to set up the DMA for a next Message Buffer Read.
6. After DMA Channel 5 completed the two data transfers, DMA Channel 06 is triggered. This DMA Channel transfers an 8 bit semaphore from SCU_SMADAT02.[15:8] to the MLI pipe 0 window.

Figure 4-6 sketches the DMA operation running autonomously after Host Controller has requested a message buffer by a write operation to the Output Command Register.

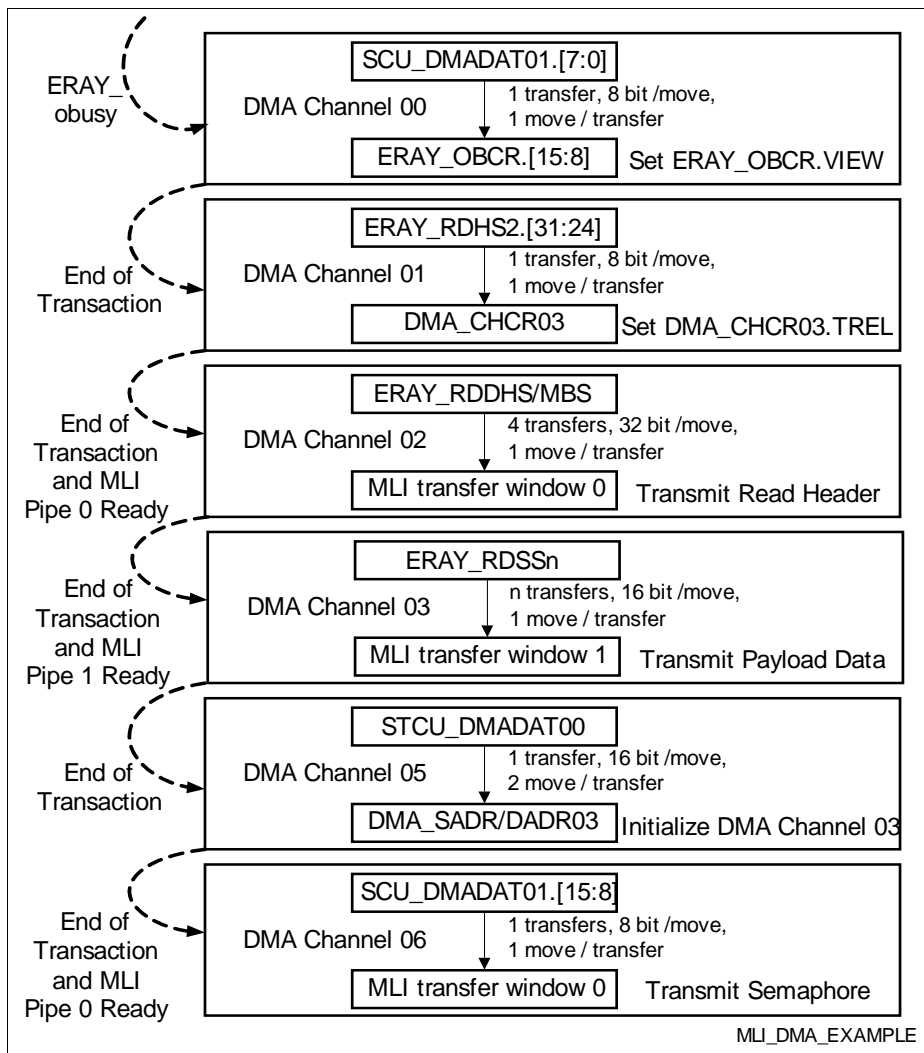


Figure 4-6 DMA Operation for Message Buffer Read Support

Table 4-4 Set-up of SAK-CIC310-OSMX2HT for Single Message Buffer Read

Register		Value to be initialized	Description
DMA_CHCR00		0018 C000 _H	The channel control register for DMA channel 00 contains its configuration and its controls: <ul style="list-style-type: none">• No Transfer Count Reload.• The E-RAY Transfer Message RAM to Output Buffer RAM busy signal is selected as trigger.• 1 move to be done for each transaction request.• Reset of TRSR.CH00 each time TCOUNT=0.• Continuous Mode operation is selected.• 8-bit (byte) data width selected.• No pattern detection.• Low Channel Priority.
TREL	00 _H		
PRSEL	110 _B		
BLKM	000 _B		
ROAT	1 _B		
CHMODE	1 _B		
CHDW	00 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR00		0000 0008 _H	The channel interrupt control register control the interrupt generation. <ul style="list-style-type: none">• Wrap source buffer interrupt disabled• Wrap destination buffer interrupt disabled• An interrupt is generated each time TCOUNT equals IRDV (0)• SR0 selected for channel 00 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0000 _B		
IRDV	0000 _B		
DMA_ADRCR00		0000 0000 _H	The address control register control the way the address is modified after each move. <ul style="list-style-type: none">• After each move operation. the source address is not modified (modification factor 1, decrement, no update of SADR[31:0])• After each move operation. the destination address is not modified (modification factor 1, decrement, no update of DADR[31:0])• Shadow Register unused.
SMF	000 _B		
INCS	0 _B		
DMF	000 _B		
INCD	0 _B		
CBLS	0000 _B		
CBLD	0000 _B		
SHCT	00 _B		
DMA_SADR00		0000 0884 _H	The source address register contains the 32-bit address of the data source. <ul style="list-style-type: none">• Source Address is SCU_DMADAT01
SADR	0000 0884 _H		
DMA_DADR00		0000 1715 _H	The destination address register contains the 32-bit address of the data destination. <ul style="list-style-type: none">• Destination Address is ERAY_OBCR.[15:8]
SADR	0000 1715 _H		

Host Communication Channels

Table 4-4 Set-up of SAK-CIC310-OSMX2HT for Single Message Buffer Read

Register		Value to be initialized	Description
DMA_CHCR01		0018 0000 _H	<p>The channel control register for DMA channel 01 contains its configuration and its controls:</p> <ul style="list-style-type: none"> • No Transfer Count Reload. • The DMA_ch00_out is selected as trigger. • 1 move to be done for each transaction request. • Reset of TRSR.CH01 each time TCOUNT=0. • Continuous Mode operation is selected. • 8-bit (byte) data width selected. • No pattern detection. • Low Channel Priority.
TREL	00 _H		
PRSEL	000 _B		
BLKM	000 _B		
ROAT	1 _B		
CHMODE	1 _B		
CHDW	00 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR01		0000 0108 _H	<p>The channel interrupt control register control the interrupt generation.</p> <ul style="list-style-type: none"> • Wrap source buffer interrupt disabled • Wrap destination buffer interrupt disabled • An interrupt is generated each time TCOUNT equals IRDV (0) • SR1 selected for channel 01 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0001 _B		
IRDV	0000 _B		
DMA_ADRCR01		0000 0000 _H	<p>The address control register control the way the address is modified after each move.</p> <ul style="list-style-type: none"> • After each move operation. the source address is not modified (modification factor 1, decrement, no update of DADR[31:0]) • After each move operation. the destination address is not modified (modification factor 1, decrement, no update of DADR[31:0]) • Shadow Register unused.
SMF	000 _B		
INCS	0 _B		
DMF	000 _B		
INCD	0 _B		
CBLS	0000 _B		
CBLD	0000 _B		
SHCT	00 _B		
DMA_SADR01		0000 1707 _H	<p>The source address register contains the 32-bit address of the data source.</p> <ul style="list-style-type: none"> • Source Address is ERAY_RDHS2.PLR
SADR	0000 1707 _H		
DMA_DADR01		0000 04E4 _H	<p>The destination address register contains the 32-bit address of the data destination.</p> <ul style="list-style-type: none"> • Destination Address is DMA_CHCR03.[15:0]
SADR	0000 04E4 _H		

Host Communication Channels
Table 4-4 Set-up of SAK-CIC310-OSMX2HT for Single Message Buffer Read

Register		Value to be initialized	Description
DMA_CHCR02		0050 8004 _H	The channel control register for DMA channel 02 contains its configuration and its controls: <ul style="list-style-type: none"> • Transfer Count Reloaded with 4. • The Ready Flag RF is selected as trigger. • 4 move to be done for each transaction request. • Reset of TRSR.CH02 after each transfer. • Continuous Mode operation is selected. • 32-bit (word) data width selected. • No pattern detection. • Low Channel Priority.
TREL	04 _H		
PRSEL	100 _B		
BLKM	000 _B		
ROAT	0 _B		
CHMODE	1 _B		
CHDW	10 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR02		0000 0208 _H	The channel interrupt control register control the interrupt generation. <ul style="list-style-type: none"> • Wrap source buffer interrupt disabled • Wrap destination buffer interrupt disabled • An interrupt is generated each time TCOUNT equals IRDV (0) • SR2 selected for channel 02 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0010 _B		
IRDV	0000 _B		
DMA_ADRCR02		0000 4488 _H	The address control register control the way the address is modified after each move. <ul style="list-style-type: none"> • After each move operation. the source address is incremented by 4 (word data width) using a circular addressing of 16 Byte circular buffer length (no update of SADR[31:4]). • After each move operation. the source address is incremented by 4 (word data width) using a circular addressing of 16 Byte circular buffer length (no update of DADR[31:4]). • Shadow Register unused.
SMF	000 _B		
INCS	1 _B		
DMF	000 _B		
INCD	1 _B		
CBLS	0100 _B		
CBLD	0100 _B		
SHCT	00 _B		
DMA_SADR02		0000 1700 _H	The source address register contains the 32-bit start address of the source buffer. <ul style="list-style-type: none"> • Start Address is ERAY_RDHS1
SADR	0000 1700 _H		

Host Communication Channels

Table 4-4 Set-up of SAK-CIC310-OSMX2HT for Single Message Buffer Read

Register		Value to be initialized	Description
DMA_DADR02		0000 8100 _H	The destination address register contains the 32-bit start address of the destination buffer. <ul style="list-style-type: none"> Start Address is the Small Transfer Windows (8 Kbyte max.) of Pipe 0, offset 100_H (to be stored after the payload data).
DADR	0000 8100 _H		
DMA_CHCR03		0030 8000 _H	The channel control register for DMA channel 03 contains its configuration and its controls: <ul style="list-style-type: none"> No Transfer Count Reload. The Ready Flag RF is selected as trigger. 1 move to be done for each transaction request. Reset of TRSR.CH03 after each transfer. Continuous Mode operation is selected. 16-bit (half word) data width selected. No pattern detection. Low Channel Priority.
TREL	00 _H		
PRSEL	100 _B		
BLKM	000 _B		
ROAT	0 _B		
CHMODE	1 _B		
CHDW	01 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR03		0000 0308 _H	The channel interrupt control register control the interrupt generation. <ul style="list-style-type: none"> Wrap source buffer interrupt disabled Wrap destination buffer interrupt disabled An interrupt is generated each time TCOUNT equals IRDV (0) SR3 selected for channel 03 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0011 _B		
IRDV	0000 _B		
DMA_ADRCR03		0000 8888 _H	The address control register control the way the address is modified after each move. <ul style="list-style-type: none"> After each move operation. the source address is incremented by 2 (half word data width) using a circular addressing of 256 Byte circular buffer length (no update of SADR[31:8]). After each move operation. the source address is incremented by 2 (half word data width) using a circular addressing of 256 Byte circular buffer length (no update of DADR[31:8]). Shadow Register unused.
SMF	000 _B		
INCS	1 _B		
DMF	000 _B		
INCD	1 _B		
CBLS	1000 _B		
CBLD	1000 _B		
SHCT	00 _B		

Host Communication Channels

Table 4-4 Set-up of SAK-CIC310-OSMX2HT for Single Message Buffer Read

Register		Value to be initialized	Description
DMA_SADR03		0000 1600 _H	The source address register contains the 32-bit start address of the source buffer. <ul style="list-style-type: none"> Start Address is ERAY_RDDS01
SADR		0000 1600 _H	
DMA_DADR03		0000 A000 _H	The destination address register contains the 32-bit start address of the destination buffer. <ul style="list-style-type: none"> Start Address is the Small Transfer Windows (8 Kbyte max.) of Pipe 1.
DADR		0000 A000 _H	
DMA_CHICR04		0000 0408 _H	The channel interrupt control register control the interrupt generation. <ul style="list-style-type: none"> Wrap source buffer interrupt disabled Wrap destination buffer interrupt disabled An interrupt is generated each time TCOUNT equals IRDV (0) SR4 selected for channel 04 interrupt
WRPSE		0 _B	
WRPDE		0 _B	
INTCT		10 _B	
WRPP		0000 _B	
INTP		0100 _B	
IRDV		0000 _B	
DMA_CHCR05		0059 2000 _H	The channel control register for DMA channel 05 contains its configuration and its controls: <ul style="list-style-type: none"> No Transfer Count Reload. The Ready Flag RF is selected as trigger. 2 move to be done for each transaction request. Reset of TRSR.CH05 each time TCOUNT=0. Continuous Mode operation is selected. 16-bit (word) data width selected. No pattern detection. Low Channel Priority.
TREL		00 _H	
PRSEL		001 _B	
BLKM		001 _B	
ROAT		1 _B	
CHMODE		1 _B	
CHDW		01 _B	
PATSEL		00 _B	
CHPRIO		0 _B	
DMA_CHICR05		0000 0508 _H	The channel interrupt control register control the interrupt generation. <ul style="list-style-type: none"> Wrap source buffer interrupt disabled Wrap destination buffer interrupt disabled An interrupt is generated each time TCOUNT equals IRDV (0) SR5 selected for channel 05 interrupt
WRPSE		0 _B	
WRPDE		0 _B	
INTCT		10 _B	
WRPP		0000 _B	
INTP		0101 _B	
IRDV		0000 _B	

Host Communication Channels
Table 4-4 Set-up of SAK-CIC310-OSMX2HT for Single Message Buffer Read

Register		Value to be initialized	Description
DMA_ADRCR05		0000 3388 _H	The address control register control the way the address is modified after each move. <ul style="list-style-type: none"> After each move operation. the source address is incremented by 2 (half word data width) using a circular addressing of 4 Byte circular buffer length (no update of SADR[31:2]). After each move operation. the source address is incremented by 4 (word data width) using a circular addressing of 8 Byte circular buffer length (no update of DADR[31:3]). Shadow Register unused.
SMF	000 _B		
INCS	1 _B		
DMF	001 _B		
INCD	1 _B		
CBLS	0010 _B		
CBLD	0011 _B		
SHCT	00 _B		
DMA_SADR05		0000 0880 _H	The source address register contains the 32-bit start address of the source buffer. <ul style="list-style-type: none"> Start Address is STCU_DMADAT00
SADR	0000 0880 _H		
DMA_DADR05		0000 04D0 _H	The destination address register contains the 32-bit start address of the destination buffer. <ul style="list-style-type: none"> Start Address is DMA_SADR02.
DADR	0000 04D0 _H		
DMA_CHCR06		0018 8000 _H	The channel control register for DMA channel 06 contains its configuration and its controls: <ul style="list-style-type: none"> No Transfer Count Reload. The Ready Flag RF signal is selected as trigger. 1 move to be done for each transaction request. Reset of TRSR.CH06 each time TCOUNT=0. Continuous Mode operation is selected. 8-bit (byte) data width selected. No pattern detection. Low Channel Priority.
TREL	00 _H		
PRSEL	001 _B		
BLKM	000 _B		
ROAT	1 _B		
CHMODE	1 _B		
CHDW	00 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR06		0000 0608 _H	The channel interrupt control register control the interrupt generation. <ul style="list-style-type: none"> Wrap source buffer interrupt disabled Wrap destination buffer interrupt disabled An interrupt is generated each time TCOUNT equals IRDV (0) SR6 selected for channel 06 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0110 _B		
IRDV	0000 _B		

Host Communication Channels

Table 4-4 Set-up of SAK-CIC310-OSMX2HT for Single Message Buffer Read

Register		Value to be initialized	Description
DMA_ADRCR06		0000 0000 _H	<p>The address control register control the way the address is modified after each move.</p> <ul style="list-style-type: none">After each move operation. the source address is not modified (modification factor 1, decrement, no update of SADR[31:0])After each move operation. the destination address is not modified (modification factor 1, decrement, no update of DADR[31:0])Shadow Register unused.
SMF	000 _B		
INCS	0 _B		
DMF	000 _B		
INCD	0 _B		
CBLS	0000 _B		
CBLD	0000 _B		
SHCT	00 _B		
DMA_SADR06		0000 0885 _H	<p>The source address register contains the 32-bit start address of the source buffer.</p> <ul style="list-style-type: none">Start Address is STCU_DMADAT01.[15:8]
SADR	0000 0885 _H		
DMA_DADR06		0000 8110 _H	<p>The destination address register contains the 32-bit address of the data destination.</p> <ul style="list-style-type: none">Destination Address is the Small Transfer Window (8 Kbyte max.) of Pipe 0, offset 110_H (to be stored after the payload data).
SADR	0000 8110 _H		
DMA_CHICR07		0000 0708 _H	<p>The channel interrupt control register control the interrupt generation.</p> <ul style="list-style-type: none">Wrap source buffer interrupt disabledWrap destination buffer interrupt disabledAn interrupt is generated each time TCOUNT equals IRDV (0)SR7 selected for channel 07 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0111 _B		
IRDV	0000 _B		

Host Communication Channels

Table 4-4 Set-up of SAK-CIC310-OSMX2HT for Single Message Buffer Read

Register		Value to be initialized	Description
DMA_HTREQ		0000 006F _H	Enable Hardware Transfer Request for DMA Channel 00 through DMA Channel 03 and DMA Channel 05 through DMA Channel 06.
ECH00	1 _B		
ECH01	1 _B		
ECH02	1 _B		
ECH03	1 _B		
ECH04	0 _B		
ECH05	1 _B		
ECH06	1 _B		
ECH07	0 _B		
ERAY_CUST1		0000 0006 _H	The Busy Control Register configures the automatic delay scheme. <ul style="list-style-type: none"> Enable auto delay scheme for Output and Input Buffer Control Register (OBCR and IBCR)
INT0	0 _B		
OEN	1 _B		
IEN	1 _B		
ERAY_CUST3		0000 02FF _H	The Time-out Counter Register configures the time-out counter reload value for the automatic delay scheme.
T0	0000 02FF _H		
ERAY_OBCM		0000 0003 _H	The Output Buffer Command Mask configures how the Output Buffer is updated from the message buffer in the Message RAM. <ul style="list-style-type: none"> Header and Data is transfer from Message RAM to Output Buffer
RHSS	1 _B		
RDSS	1 _B		
SCU_DMADAT00		A000 1600 _H	[15:0] Address of ERAY_RDDS01. [31:16]Address of MLI Small Transfer Window Pipe 1 for DMA_DADR03 for Data Transfer.
DDAT	A000 1600 _H		
SCU_DMADAT01		0000 0101 _H	[7:0] Data for ERAY_OBCR to toggle between E-Ray Shadow and Host Output Buffer [15:8] Semaphore to communicate a completed transfer to Host Controller.
DDAT	0000 0101 _H		
SCU_CHTR02		0000 0006 _H	The bit field TRSEL defines the trigger source for the DMA channel 02. Signal dma_ch01_o is selected as trigger source of DMA Channel 02
TRSEL	0000 0006 _H		

Host Communication Channels

Table 4-4 Set-up of SAK-CIC310-OSMX2HT for Single Message Buffer Read

Register		Value to be initialized	Description
SCU_CHTR03		0000 0006 _H	The bit field TRSEL defines the trigger source for the DMA channel 03. Signal dma_ch02_o is selected as trigger source of DMA Channel 03
TRSEL		0000 0006 _H	
SCU_CHTR06		0000 0006 _H	The bit field TRSEL defines the trigger source for the DMA channel 06. Signal dma_ch05_o is selected as trigger source of DMA Channel 06
TRSEL		0000 0006 _H	
MLI_TIER		0xxx xxx3 _H	The transmitter interrupt enable register enables/disables the interrupts generated by the MLI transmitter. <ul style="list-style-type: none"> Normal Frame Sent in Pipe 0 and 1 Interrupt is Enabled. Therefore RF Flip-Flop is set after every completed MLI Pipe 0 or Pipe 1 transfer.
NFSIE0		1 _B	
NFSIE1		1 _B	
NFSIE2		x _B	
NFSIE3		x _B	
MLI_TINPR		0xxx xx10 _H	The transmitter interrupt node pointer register routes the generated interrupts to respective interrupt output lines (MLIx_O). <ul style="list-style-type: none"> MLI interrupt output 0 (MLI0_O) is selected for Normal Frame Sent in Pipe 0 Interrupt. MLI interrupt output 1 (MLI1_O) is selected for Normal Frame Sent in Pipe 1 Interrupt
NFSIP0		000 _B	
NFSIP1		001 _B	
MLI_GINTR		0000 0003 _H	The Global Interrupt Set Register can activate the interrupt output lines of the MLI module. <ul style="list-style-type: none"> The MLI interrupt output line 0 will be activated. The MLI interrupt output line 1 will be activated. <i>Note: This register is the last to be configured, else wise the RF flip flop may not be set correctly.</i>
SIMLI0		1 _B	
SIMLI1		1 _B	
SIMLI2		0 _B	
SIMLI3		0 _B	
SIMLI4		0 _B	
SIMLI5		0 _B	
SIMLI6		0 _B	
SIMLI7		0 _B	

4.4.4.2 Periodically Copy Message Buffer Contents to Host Memory

This chapter describes the usage of the DMA to read periodically the content of a user defined selection of message buffer (up to 20) and store the content in a memory of the the host controller (content mirror). So if the Host controller has to read a respective message buffer, it only has to read the local memory (mirror buffer). Special care has to

Host Communication Channels

be taken to prevent any data access to the mirrored data area while the DMA is updating the respective memory content. Therefore the SAK-CIC310-OSMX2HT generates a hardware signal (SR7) to communicate the begin of a transfer. After the last transfer (Mirror memory completed) a hardware signal is generated to signalize the ongoing last transfer. This example allows to freely select a list of up to 20 Message buffers to be copied periodically into a respective memory of the host.

The DMA channel 00 receives a periodic trigger from the Host Controller writing a 43_H into DMA_STREQ (set a software trigger for MDA channel 00, DMA Channel 01, and DMA Channel 06), the E-Ray Timer Interrupt 0 (absolute timer in terms of cycle count and microticks offset), or a hardware trigger from the Host controller (in this case additionally to the following described configuration of SAK-CIC310-OSMX2HT the user has to initialize trig1 respectively and change the trigger condition of DMA channel 00). On Service Request Output Line 7 (SR7) a pulse is generated at the beginning of each message buffer transfer and two consecutive during the last transfer. Until the DMA signalizes the completion of the message buffer block with a double pulse on the Service Request Output Line 7 (SR7), the host may not access the mirror memory to avoid data inconsistency.

1. DMA Channel 00 is triggered by the cyclic trigger signal (in this example ERAY_TINT0 or SW Trigger) and initializes the Source Address Register of the DMA channel 06 (DMA_SADR06) and the Source address register of the DMA channel 07 (DMA_SADR07) using SCU_DMADAT11[31:16].
2. After DMA Channel 00 completed all the data moves DMA Channel 01 is triggered. DMA Channel 01 initializes the Destination Address Register of the DMA channel 04 (DMA_DADR04) by copying the respective data from SCU_DMADAT00.
3. After DMA Channel 01 completed the data move, it triggers DMA Channel 06 by driving a pulse on the external Service Request Output Line 7 (SR7) and enters the following loop.
 - a) After DMA Channel 05 completed all data moves, DMA Channel 01 completed all the data moves, or the software trigger (STREQ.SCH05) is applied, DMA Channel 06 is triggered. DMA Channel 06 initializes the DMA_SADR05 and DMA_DADR05 for the next transfer of payload data. This enables a flexible allocation of memory for the Message Buffer Mirror Memory within the Host Controller. The locations SCU_DMADAT12 through SCU_DMADAT31 contains a list of up to 20 memory locations. The higher 16 bit contain the MLI Pipe 1 Offset to store the payload data. The lower 16 bit always contains the offset of the ERAY_RDDS01 register. Take care, that the first transfer is stored as last of the list.
 - b) After DMA Channel 06 completed DMA Channel 07 is triggered. DMA Channel 06 requests the next message buffer to be copied into the memory of the host controller. The message buffer number is stored in a list of 16 bit values within the locations SCU_DMADAT01 through SCU_DMADAT10. The lower 8 bit of each 16 bit location contains the number of the message buffer. The higher 8 bit always contain a 02_H to respectively set the ERAY_OBCR.REQ bit. The end of the list is

Host Communication Channels

marked by a 0002_H end marker. In case of 20 Message Buffers to be copied, SCU_DMADAT11.[15:0] is used to store the end marker of the message buffer list. After the DMA channel 07 has completed the move operation and the E-Ray module copied the requested message buffer into the shadow output buffer, the ERAY_OBUSY signal will be cleared and the loop continues.

- c) DMA Channel 02 is triggered by this ERAY_OBUSY signal and sets the ERAY_OBCR.VIEW bit by moving a 01_H from SCU_DMADAT11.[7:0] to ERAY_OBCR.[15:8].
- d) After DMA Channel 02 completed transfer, DMA Channel 03 is triggered. DMA Channel 03 moves the received payload length of the selected message buffer into the transfer count reload bit field of DMA channel 05 (DMA_CHCR05.TREL).
- e) After DMA Channel 03 completed DMA Channel 05 is triggered as soon as the MLI Pipe 0 is ready for a transfer. DMA Channel 04 transfers four 32 Bit data from the ERAY_RDHS0, ERAY_RDHS1, ERAY_RDHS3, and ERAY_MBS registers to the MLI Pipe 0 as write transfer. After every 32 bit transfer, the DMA_Channel waits for the MLI interrupt signalling a completed previous transfer.
- f) After DMA Channel 04 completed move, DMA Channel 05 is triggered as soon as the MLI Pipe 1 is ready for a transfer. DMA Channel 04 transfers several 16 Bit data from the RDDS registers as specified in the RDHS2.PLR bit field to the MLI Pipe 1 as write transfer. After every 16 bit transfer, the DMA Channel 05 waits for the MLI interrupt signalling a completed previous transfer.
4. The DMA will exit the loop as soon as DMA channel 07 stops requesting a new message buffer (ERAY_OBCR.REQ=0) due to transferring the end marker to ERAY_OBCR register. A respective pattern detection within the move engine will recognize this end marker and signalize the end of the transfer via a hardware signal (SR 2) to the host controller. The host controller may now continue accessing the mirror memory area after the ongoing MLI transfer has completed. The data may now be accessed by the CPU. As soon as the CPU does no longer use the data, it has to enable DMA channel 7 by writing to DMA_HTREQ 0000 0080_H. The DMA is now ready to serve the next cyclic transfer signal.

Table 4-5 Content of the Host Mirror Memory

MLI Pipe 0		MLI Pipe 1	
Offset: SCU_DMADAT31.[15:8]		Offset: SCU_DMADAT00.[31:16]	
Message Object Header "A"	RDHS1	Message Object "A"	RDDS1
	RDHS2		RDDS2
	RDHS3		...
	MBS		RDDSa
Message Object Header "B"	RDHS1	Message Object "B"	RDDS1
	RDHS2		RDDS2
	RDHS3		...
	MBS		RDDSB
Message Object Header "C"	RDHS1	Message Object "C"	RDDS1
	RDHS2		RDDS2
	RDHS3		...
	MBS		RDDSc
...		...	
Message Object Header "J"	RDHS1	Message Object "J"	RDDS1
	RDHS2		RDDS2
	RDHS3		...
	MBS		RDDSJ

The memory location of a respective Message Object Header Block x (MOHx); x=[0...19] within the mirror memory area can be calculated by following formula:

$$MOH_x = HOST:MLI_RP0BAR + IFLEX:SCU_DMADAT00 + (4 \times x) \quad (4.1)$$

Host Communication Channels

The memory location of a respective Message Object Payload Data Block y (MOPD y); $y=[0\dots19]$ within the mirror memory area can be calculated by following formula:

$$\text{MOPD}_y = \text{HOST:MLI_RP1BAR} + \text{IFLEX:SCU_DMADAT}(y+12).[31:16] \quad (4.2)$$

The DMA can not check if a buffer overrun happens, so the user has to choose each MOPD y respectively.

The SCU $\overline{\text{DMADAT}}$ register are used to store the list of message buffers to be read and the location where to store the content within the host controller. Not shown here are the initialization of the MLI communication itself, especially the Transmitter Pipe 0 and 1 Address Offset Register (TP0AOFR, TP1AOFR), the Transmitter Pipe 0 and 1 Base Address Register (TP0BAR, TP1BAR), and the Receiver Pipe 0 and 1 Base Address Register (RP0BAR, RP1BAR).

Table 4-6 Content of the SCU_DMADATxx Register

MOPDy Offset List			List of Message Object Number		
DMADAT01	[15:0]	200 _H + Message Object Number of "A"	DMADAT12	[15:0]	1600 _H (RDDS01)
	[31:16]	200 _H + Message Object Number of "B"		[31:16]	Message Object "A" Offset 0000 8000 _H +Offset
DMADAT02	[15:0]	200 _H + Message Object Number of "C"	DMADAT13	[15:0]	1600 _H (RDDS01)
	[31:16]	200 _H + Message Object Number of "D"		[31:16]	Message Object "B" Offset 0000 8000 _H +Offset
DMADAT03	[15:0]	200 _H + Message Object Number of "E"	DMADAT14	[15:0]	1600 _H (RDDS01)
	[31:16]	200 _H + Message Object Number of "F"		[31:16]	Message Object "C" Offset 0000 8000 _H +Offset
...			...		
DMADAT10	[15:0]	200 _H + Message Object Number of "I"	DMADAT31	[15:0]	1600 _H (RDDS01)
	[31:16]	200 _H + Message Object Number of "J"		[31:16]	Message Object "J" Offset 0000 8000 _H +Offset
Other Data					
DMADAT00	[31:0]	Offset of Header Mirror Block Pipe 1 0000 A000 _H + Offset	DMADAT11	[7:0]	01 _H (OBCR.VIEW)
				[15:8]	00 _H
				[23:16]	84 _H (DMADAT01)
				[31:24]	B0 _H (DMADAT12)

Figure 4-7 sketches the DMA operation running autonomously on every cyclic trigger.

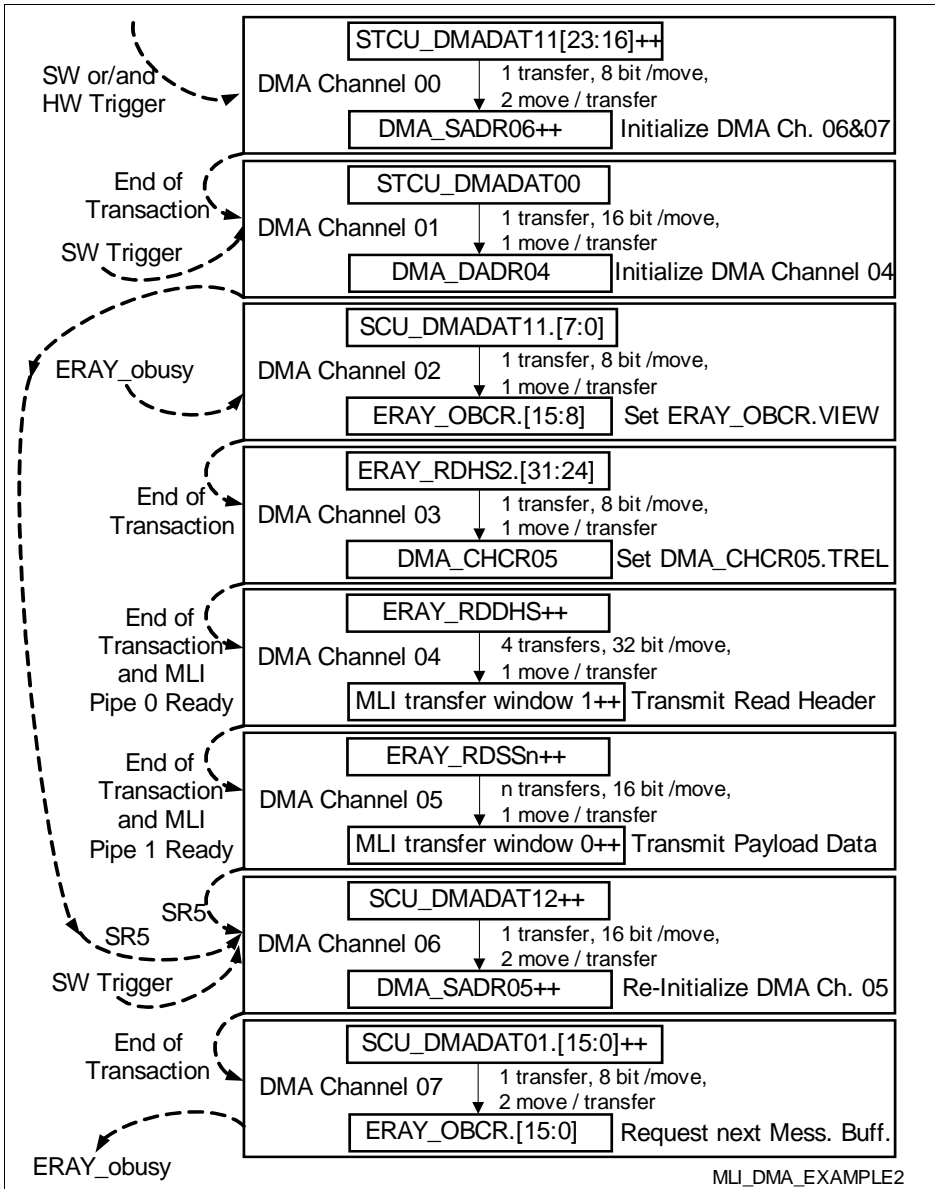


Figure 4-7 DMA Operation for Message Buffer Read Support

Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
ERAY_CUST1		0000 0006 _H	The Busy Control Register configures the automatic delay scheme. <ul style="list-style-type: none"> Enable auto delay scheme for Output and Input Buffer Control Register (OBCR and IBCR)
INT0	0 _B		
OEN	1 _B		
IEN	1 _B		
ERAY_CUST3		0000 01FF _H	The Time-out Counter Register configures the time-out counter reload value for the automatic delay scheme.
T0	0000 01FF _H		
ERAY_OBCM		0000 0003 _H	The Output Buffer Command Mask configures how the Output Buffer is updated from the message buffer in the Message RAM. <ul style="list-style-type: none"> Header and Data is transfer from Message RAM to Output Buffer
RHSS	1 _B		
RDSS	1 _B		
SCU_CHTR00		0000 0004 _H	The bit field TRSEL defines the trigger source for the DMA channel 00. E-Ray Signal TINT0 is selected as trigger source of DMA Channel 00
TRSEL	0000 0004 _H		
SCU_CHTR01		0000 0006 _H	The bit field TRSEL defines the trigger source for the DMA channel 01. Signal dma_ch00_o is selected as trigger source of DMA Channel 01
TRSEL	0000 0006 _H		
SCU_CHTR03		0000 0006 _H	The bit field TRSEL defines the trigger source for the DMA channel 03. Signal dma_ch02_o is selected as trigger source of DMA Channel 03
TRSEL	0000 0006 _H		
SCU_CHTR04		0000 0006 _H	The bit field TRSEL defines the trigger source for the DMA channel 04. Signal dma_ch03_o is selected as trigger source of DMA Channel 04
TRSEL	0000 0006 _H		
SCU_CHTR05		0000 0006 _H	The bit field TRSEL defines the trigger source for the DMA channel 05. Signal dma_ch04_o is selected as trigger source of DMA Channel 05
TRSEL	0000 0006 _H		
SCU_CHTR06		0000 0002 _H	The bit field TRSEL defines the trigger source for the DMA channel 06. The external Trigger Signal 1 is selected as trigger source of DMA Channel 06
TRSEL	0000 0002 _H		
SCU_CHTR07		0000 0006 _H	The bit field TRSEL defines the trigger source for the DMA channel 07. Signal dma_ch06_o is selected as trigger source of DMA Channel 07
TRSEL	0000 0006 _H		

Host Communication Channels

Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
SCU_SRCR		0001 xxxx _H	<p>This Register selects the internal status signals used as source for interrupt output signals (INTO0 to INTO4).</p> <ul style="list-style-type: none">Select non inverted interrupt signal source DMA_SR0 as INTO4
INSEL0	xxx _B		
INV0	x _B		
INSEL1	xxx _B		
INV1	x _B		
INSEL2	xxx _B		
INV2	x _B		
INSEL3	001 _B		
INV3	0 _B		
SCU_ETCTR		0000 xx47 _H	<p>This Register contains the bits defining the behavior of the external trigger signals TRIG0 and TRIG1.</p> <ul style="list-style-type: none">Select falling edge of external service request signal source SR7 as trig0.
INSEL0	0111 _B		
FEN0	1 _B		
REN0	0 _B		
INSEL1	xxx _B		
FEN1	x _B		
REN1	x _B		
P1_IOCRR12		xxxx xxAA _H	<p>This Register contains the bits selecting the digital output characteristics, such as port direction (input/output) and alternate output selections.</p> <ul style="list-style-type: none">Select INT_O0 (ALT2) for push-pull output.
PC12B	1010 _B		
PC12	1010 _B		
PC13B	xxx _B		
PC13	xxx _B		
PC14B	xxx _B		
PC14	xxx _B		
PC15B	xxx _B		
PC15	xxx _B		

Host Communication Channels

Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
DMA_HTREQ		0000 00FF _H	Enable Hardware Transfer Request for all DMA Channels.
ECH00	1 _B		
ECH01	1 _B		
ECH02	1 _B		
ECH03	1 _B		
ECH04	1 _B		
ECH05	1 _B		
ECH06	1 _B		
ECH07	1 _B		
DMA_MEOPR		0000 0001 _H	The move engine pattern register defines the patterns to be searched for by respective data moves. <ul style="list-style-type: none"> Search for 0001_H
PAT00	01 _H		
PAT01	00 _H		
PAT02	00 _H		
PAT03	00 _H		
DMA_CHCR00		0019 A000 _H	The channel control register for DMA channel 00 contains its configuration and its controls: <ul style="list-style-type: none"> No Transfer Count Reload. ERAY_TINT0 via Trigger Flag TF is trigger. 2 moves to be done for each transaction request. Reset of TRSR.CH00 each time TCOUNT=0 after a transfer. Continuous Mode operation is selected. 8-bit (byte) data width selected. No pattern detection. Low Channel Priority.
TREL	00 _H		
PRSEL	101 _B		
BLKM	001 _B		
ROAT	1 _B		
CHMODE	1 _B		
CHDW	00 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR00		0000 0008 _H	The channel interrupt control register control the interrupt generation. <ul style="list-style-type: none"> Wrap source buffer interrupt disabled Wrap destination buffer interrupt disabled An interrupt is generated each time TCOUNT equals IRDV (0) SR0 selected for channel 00 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0000 _B		
IRDV	0000 _B		

Host Communication Channels

Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
DMA_ADRCR00		0000 61D8 _H	<p>The address control register control the way the address is modified after each move.</p> <ul style="list-style-type: none">After each move operation. the source address is incremented by 1 (modification factor 1, increment, no update of SADR[31:1])After each move the destination address is incremented by 32 (modification factor 32, increment, no update of DADR[31:6])Shadow Register unused.
SMF	000 _B		
INCS	1 _B		
DMF	101 _B		
INCD	1 _B		
CBLS	0001 _B		
CBLD	0110 _B		
SHCT	00 _B		
DMA_SADR00		0000 08FA _H	<p>The source address register contains the two 32-bit address of the data source.</p> <ul style="list-style-type: none">Source Address is SCU_DMADAT30.[23:16]
SADR	0000 08FE _H		
DMA_DADR00		0000 0550 _H	<p>The destination address register contains the 32-bit address of the data destination.</p> <ul style="list-style-type: none">Destination Address is DMA_SADR06
SADR	0000 0550 _H		
DMA_CHCR01		0058 A000 _H	<p>The channel control register for DMA channel 01 contains its configuration and its controls:</p> <ul style="list-style-type: none">No Transfer Count Reload.The DMA_ch00_out via Trigger Flag TF is as trigger.1 move to be done for each transaction request.Reset of TRSR.CH01 each time TCOUNT=0.Continuous Mode operation is selected.32-bit (word) data width selected.No pattern detection.Low Channel Priority.
TREL	00 _H		
PRSEL	101 _B		
BLKM	000 _B		
ROAT	1 _B		
CHMODE	1 _B		
CHDW	10 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR01		0000 0108 _H	<p>The channel interrupt control register control the interrupt generation.</p> <ul style="list-style-type: none">Wrap source buffer interrupt disabledWrap destination buffer interrupt disabledAn interrupt is generated each time TCOUNT equals IRDV (0)SR5 selected for channel 01 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0101 _B		
IRDV	0000 _B		

Host Communication Channels

Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
DMA_ADRCR01		0000 0000 _H	<p>The address control register control the way the address is modified after each move.</p> <ul style="list-style-type: none">After each move operation. the source address is not modified (modification factor 1, decrement, no update of SADR[31:0])After each move operation. the destination address is not modified (modification factor 1, decrement, no update of DADR[31:0])Shadow Register unused.
SMF	000 _B		
INCS	0 _B		
DMF	000 _B		
INCD	0 _B		
CBLS	0000 _B		
CBLD	0000 _B		
SHCT	00 _B		
DMA_SADR01		0000 0880 _H	<p>The source address register contains the 32-bit address of the data source.</p> <ul style="list-style-type: none">Source Address is SCU_DMADAT00
SADR	0000 0880 _H		
DMA_DADR01		0000 0514 _H	<p>The destination address register contains the 32-bit address of the data destination.</p> <ul style="list-style-type: none">Destination Address is DMA_DADR04
SADR	0000 0514 _H		
DMA_CHCR02		0018 C000 _H	<p>The channel control register for DMA channel 02 contains its configuration and its controls:</p> <ul style="list-style-type: none">No Transfer Count Reload.The E-RAY Transfer Message RAM to Output Buffer RAM busy signal is selected as trigger.1 move to be done for each transaction request.Reset of TRSR.CH02 each time TCOUNT=0.Continuous Mode operation is selected.8-bit (byte) data width selected.No pattern detection.Low Channel Priority.
TREL	00 _H		
PRSEL	110 _B		
BLKM	000 _B		
ROAT	1 _B		
CHMODE	1 _B		
CHDW	00 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR02		0000 0208 _H	<p>The channel interrupt control register control the interrupt generation.</p> <ul style="list-style-type: none">Wrap source buffer interrupt disabledWrap destination buffer interrupt disabledAn interrupt is generated each time TCOUNT equals IRDV (0)SR2 selected for channel 02 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0010 _B		
IRDV	0000 _B		

Host Communication Channels

Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
DMA_ADRCR02		0000 0000 _H	<p>The address control register control the way the address is modified after each move.</p> <ul style="list-style-type: none">After each move operation. the source address is not modified (modification factor 1, decrement, no update of SADR[31:0])After each move operation. the destination address is not modified (modification factor 1, decrement, no update of DADR[31:0])Shadow Register unused.
SMF	000 _B		
INCS	0 _B		
DMF	000 _B		
INCD	0 _B		
CBLS	0000 _B		
CBLD	0000 _B		
SHCT	00 _B		
DMA_SADR02		0000 08AC _H	<p>The source address register contains the 32-bit start address of the data source.</p> <ul style="list-style-type: none">Source Address is SCU_DMADAT11
SADR	0000 08AC _H		
DMA_DADR02		0000 1715 _H	<p>The destination address register contains the 32-bit address of the data destination.</p> <ul style="list-style-type: none">Destination Address is ERAY_OBCR.[15:8].
DADR	0000 1715 _H		
DMA_CHCR03		0018 A000 _H	<p>The channel control register for DMA channel 03 contains its configuration and its controls:</p> <ul style="list-style-type: none">No Transfer Count Reload.The DMA_ch02_out via Trigger Flag TF is selected as trigger.1 move to be done for each transaction request.Reset of TRSR.CH03 each time TCOUNT=0.Continuous Mode operation is selected.8-bit (byte) data width selected.No pattern detectionLow Channel Priority.
TREL	00 _H		
PRSEL	101 _B		
BLKM	000 _B		
ROAT	1 _B		
CHMODE	1 _B		
CHDW	00 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR03		0000 0308 _H	<p>The channel interrupt control register control the interrupt generation.</p> <ul style="list-style-type: none">Wrap source buffer interrupt disabledWrap destination buffer interrupt disabledAn interrupt is generated each time TCOUNT equals IRDV (0)SR3 selected for channel 03 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0011 _B		
IRDV	0000 _B		

Host Communication Channels
Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
DMA_ADRCR03		0000 0000 _H	<p>The address control register control the way the address is modified after each move.</p> <ul style="list-style-type: none">After each move operation. the source address is not modified (modification factor 1, decrement, no update of SADR[31:0])After each move operation. the destination address is not modified (modification factor 1, decrement, no update of DADR[31:0])Shadow Register unused.
SMF	000 _B		
INCS	0 _B		
DMF	000 _B		
INCD	0 _B		
CBLS	0000 _B		
CBLD	0000 _B		
SHCT	00 _B		
DMA_SADR03		0000 1707 _H	<p>The source address register contains the 32-bit address of the data source.</p> <ul style="list-style-type: none">Source Address is ERAY_RDHS2.PLR
SADR	0000 1707 _H		
DMA_DADR03		0000 0524 _H	<p>The destination address register contains the 32-bit address of the data destination.</p> <ul style="list-style-type: none">Destination Address is DMA_CHCR05.[15:0]
SADR	0000 0524 _H		
DMA_CHCR04		0050 8004 _H	<p>The channel control register for DMA channel 04 contains its configuration and its controls:</p> <ul style="list-style-type: none">Transfer Count Reloaded with 4.The Ready Flag RF is selected as trigger.4 move to be done for each transaction request.Reset of TRSR.CH04 after each transfer.Continuous Mode operation is selected.32-bit (word) data width selected.No pattern detection.Low Channel Priority.
TREL	04 _H		
PRSEL	100 _B		
BLKM	000 _B		
ROAT	0 _B		
CHMODE	1 _B		
CHDW	10 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR04		0000 0408 _H	<p>The channel interrupt control register control the interrupt generation.</p> <ul style="list-style-type: none">Wrap source buffer interrupt disabledWrap destination buffer interrupt disabledAn interrupt is generated each time TCOUNT equals IRDV (0)SR4 selected for channel 04 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0100 _B		
IRDV	0000 _B		

Host Communication Channels

Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
DMA_ADRCR04		0000 9488 _H	<p>The address control register control the way the address is modified after each move.</p> <ul style="list-style-type: none">After each move operation. the source address is incremented by 4 (modification factor 1, increment, no update of SADR[31:4] = 16 Byte circular buffer)After each move operation. the source address is incremented by 4 (modification factor 1, increment, no update of DADR[31:9] = 512 Byte circular buffer, so enough space to store 20 Header information).Shadow Register unused.
SMF	000 _B		
INCS	1 _B		
DMF	000 _B		
INCD	1 _B		
CBLS	0100 _B		
CBLD	1001 _B		
SHCT	00 _B		
DMA_SADR04		0000 1700 _H	<p>The source address register contains the 32-bit start address of the source buffer. Start Address is ERAY_RDHS1</p>
SADR	0000 1700 _H		
DMA_CHCR05		0030 8000 _H	<p>The channel control register for DMA channel 05 contains its configuration and its controls:</p> <ul style="list-style-type: none">No Transfer Count Reload.The Ready Flag RF is selected as trigger.1 move to be done for each transaction request.Reset of TRSR.CH05 after each transfer.Continuous Mode operation is selected.16-bit (half word) data width selected.No pattern detection.Low Channel Priority.
TREL	00 _H		
PRSEL	100 _B		
BLKM	000 _B		
ROAT	0 _B		
CHMODE	1 _B		
CHDW	01 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR05		0000 0508 _H	<p>The channel interrupt control register control the interrupt generation.</p> <ul style="list-style-type: none">Wrap source buffer interrupt disabledWrap destination buffer interrupt disabledAn interrupt is generated each time TCOUNT equals IRDV (0)SR5 selected for channel 05 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0101 _B		
IRDV	0000 _B		

Host Communication Channels

Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
DMA_ADRCR05		0000 D888 _H	<p>The address control register control the way the address is modified after each move.</p> <ul style="list-style-type: none">After each move operation. the source address is incremented by 2 (modification factor 1, increment, no update of SADR[31:8] = 256 Byte circular buffer)After each move operation. the source address is incremented by 2 (modification factor 1, increment, no update of DADR[31:13] = 8 Kbyte circular buffer to avoid leaving the MLI Window.Shadow Register unused.
SMF	000 _B		
INCS	1 _B		
DMF	000 _B		
INCD	1 _B		
CBLS	1000 _B		
CBLD	1101 _B		
SHCT	00 _B		
DMA_CHCR06		0038 A000 _H	<p>The channel control register for DMA channel 06 contains its configuration and its controls:</p> <ul style="list-style-type: none">No Transfer Count Reload.External Interrupt 0 via Trigger Flag TF is enabled as trigger.2 move to be done for each transaction request.Reset of TRSR.CH06 each time TCOUNT=0.Continuous Mode operation is selected.16-bit (word) data width selected.No pattern detection.Low Channel Priority.
TREL	00 _H		
PRSEL	101 _B		
BLKM	001 _B		
ROAT	1 _B		
CHMODE	1 _B		
CHDW	01 _B		
PATSEL	00 _B		
CHPRIO	0 _B		
DMA_CHICR06		0000 0608 _H	<p>The channel interrupt control register control the interrupt generation.</p> <ul style="list-style-type: none">Wrap source buffer interrupt disabledWrap destination buffer interrupt disabledAn interrupt is generated each time TCOUNT equals IRDV (0)SR6 selected for channel 06 interrupt
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	10 _B		
WRPP	0000 _B		
INTP	0101 _B		
IRDV	0000 _B		

Host Communication Channels

Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
DMA_ADRCR06		0000 3698 _H	The address control register control the way the address is modified after each move. <ul style="list-style-type: none"> After each move operation. the source address is incremented by 2 (modification factor 1, increment, no update of SADR[31:6]) After each move operation. the source address is incremented by 4 (modification factor 2, increment, no update of DADR[31:3]) Shadow Register unused.
SMF	000 _B		
INCS	1 _B		
DMF	001 _B		
INCD	1 _B		
CBLS	0110 _B		
CBLD	0011 _B		
SHCT	00 _B		
DMA_SADR06		0000 8B0 _H	The source address register contains the 32-bit address of the data source. <ul style="list-style-type: none"> Source Address is SCU_DMADAT12
SADR	0000 08B0 _H		
DMA_DADR06		0000 0530 _H	The destination address register contains the 32-bit address of the data destination. <ul style="list-style-type: none"> Destination Address is DMA_SADR05
DADR	0000 0530 _H		

Host Communication Channels

Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
DMA_CHCR07		0119 A000 _H	The channel control register for DMA channel 07 contains its configuration and its controls: <ul style="list-style-type: none"> • No Transfer Count Reload. • The DMA_ch06_out via Trigger Flag TF is selected as trigger. • 2 move to be done for each transaction request. • Reset of TRSR.CH06 each time TCOUNT=0. • Continuous Mode operation is selected. • 8-bit (byte) data width selected. • Aligned Pattern Compare and generate interrupt. • Low Channel Priority.
TREL	00 _H		
PRSEL	101 _B		
BLKM	001 _B		
ROAT	1 _B		
CHMODE	1 _B		
CHDW	00 _B		
PATSEL	01 _B		
CHPRIO	0 _B		
DMA_CHICR07		0000 0000 _H	The channel interrupt control register control the interrupt generation. <ul style="list-style-type: none"> • Wrap source buffer interrupt disabled. • Wrap destination buffer interrupt disabled. • Only pattern detection generates interrupt. • SR0 selected for channel 07 interrupt.
WRPSE	0 _B		
WRPDE	0 _B		
INTCT	00 _B		
WRPP	0000 _B		
INTP	0000 _B		
IRDV	0000 _B		
DMA_ADRCR07		0000 0608 _H	The address control register control the way the address is modified after each move. <ul style="list-style-type: none"> • After each move operation. the source address is incremented by 4 (modification factor 1, increment, no update of SADR[31:6]) • After each move operation. the destination address is not modified (modification factor 1, decrement, no update of DADR[31:0]) • Shadow Register unused.
SMF	000 _B		
INCS	1 _B		
DMF	000 _B		
INCD	0 _B		
CBLS	0110 _B		
CBLD	0000 _B		
SHCT	00 _B		
DMA_SADR07		0000 880 _H	The source address register contains the 32-bit address of the data source. <ul style="list-style-type: none"> • Source Address is SCU_DMADAT01
SADR	0000 0880 _H		
DMA_DADR07		0000 1714 _H	The destination address register contains the 32-bit address of the data destination. <ul style="list-style-type: none"> • Destination Address is ERAY_OBCR
DADR	0000 1714 _H		

Host Communication Channels

Table 4-7 Set-up of SAK-CIC310-OSMX2HT for complete Message Buffer Read

Register		Value to be initialized	Description
MLI_TIER		0xxx xxx3 _H	The transmitter interrupt enable register enables/disables the interrupts generated by the MLI transmitter. <ul style="list-style-type: none"> Normal Frame Sent in Pipe 0 and 1 Interrupt is Enabled. Therefore RF Flip-Flop is set after every completed MLI Pipe 0 or Pipe 1 transfer.
NFSIE0	1 _B		
NFSIE1	1 _B		
NFSIE2	x _B		
NFSIE3	x _B		
MLI_TINPR		0xxx xx01 _H	The transmitter interrupt node pointer register routes the generated interrupts to respective interrupt output lines (MLIx_O). <ul style="list-style-type: none"> MLI interrupt output 1 (MLI1_O) is selected for Normal Frame Sent in Pipe 0 Interrupt. MLI interrupt output 0 (MLI0_O) is selected for Normal Frame Sent in Pipe 1 Interrupt
NFSIP0	001 _B		
NFSIP1	000 _B		
MLI_GINTR		0000 0003 _H	The Global Interrupt Set Register can activate the interrupt output lines of the MLI module. <ul style="list-style-type: none"> The MLI interrupt output line 0 will be activated. The MLI interrupt output line 1 will be activated. <i>Note: This register is the last to be configured, else wise the RF flip flop may not be set correctly.</i>
SIMLI0	1 _B		
SIMLI1	1 _B		
SIMLI2	0 _B		
SIMLI3	0 _B		
SIMLI4	0 _B		
SIMLI5	0 _B		
SIMLI6	0 _B		
SIMLI7	0 _B		
SIMLI7	0 _B		

4.5 Communication Principles SSC

The Standalone SAK-CIC310-OSMX2HT SSC communication channel is e.g. compatible with the SSC interface of the C16x, XC16x, and TriCore AUDDO microcontroller family and e.g. the ASC of the TriCore AUDDO microcontroller family. The SSC is internally configured by the Initialization Move Engine (IME) for:

- The device is configured to run in SSC Slave Mode.
- The transfer data width is 16 bit.
- The MSB is transferred and received first.
- The receiving data is latched on trailing clock edge and shifted within the shift register on leading edge.

Host Communication Channels

- The idle clock line is configured to be high.
- The leading clock edge is configured to be a high-to-low transition.
- Transmit error, receive error, and phase error are activated.
- In case of communication to an ASC communication controller, the baud rate error has to be ignored. Therefore proposal is to configure the SSC to ignore this error. Customer may activate error within the application software.
- The P1.4 port pin is configured as SSC slave select input signal (SSC1_SLSI1). Therefore the pull-up device is assigned.
- The Slave Mode transmit idle polarity is a high level, so driving the MRST to high (1) when SSC is deselected in slave mode.
- The SLSI1 is selected as Slave Mode slave select (SLS) input.
- The SCLKB is selected as Slave Mode clock input.
- The MTSRB is selected as Slave Mode receive input.
- The MRSTA is selected as Master Mode Receive Input.
- The P1.2 port pin is configured as SSC clock input signal (SSC_SCLKB). Therefore the pull-up device is assigned
- The P6.4 port pin is configured as SSC master transmit slave receive input signal (SSC_MTSRB). Therefore the pull-up device is assigned.
- The P1.5 port pin is configured as SSC master receive slave transmit output signal (SSC_MRSTB). Therefore the push/pull function is activated and the output driver characteristic set to strong driver, sharp edge.

The reset value of the baud rate generator BR is 0063_H generating a clock signal of 200 kHz at the SCLK pin if the SAK-CIC310-OSMX2HT device is driven by a 40 MHz clock source. The baud rate generator is clocked with the module clock f_{SYS} .

The required value of the Baud Rate Timer Reload Register BR for a specific baud rate can be calculated as unsigned 16-bit integer value according to the following equation:

$$\langle BR \rangle = \left(\frac{f_{SYS}}{2 \cdot \text{Baudrate}_{SSC}} \right) - 1 \quad (4.3)$$

Note: The value of a valid BR has to be > 0.

The maximum baud rate for the SSC running in slave mode is limited to $f_{SYS}/4$ with $BR=01_{H}$, which leads to a maximum baud rate of 20 MBaud (@ 80 MHz system frequency f_{SYS}).

The Standalone SAK-CIC310-OSMX2HT device can be easily connected to an external host device via a serial channel. This mode is selected by the pin $MODE0=1$ at the rising edge of the RESET signal. Operates in SSC Slave Mode the on-chip SSC can be connected to an external SSC in Master Mode according to [Figure 4-8](#). The RDY output signals the availability of the SSC Slave. This signal can be used to detect overload scenario of the SSC slave device.

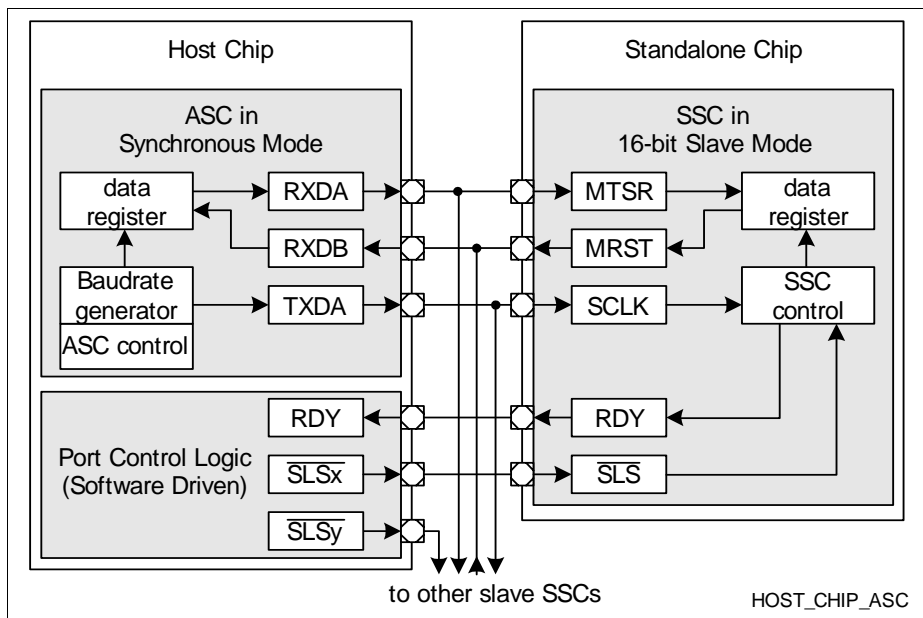


Figure 4-8 SSC in Slave Mode Communicating to an ASC

The Standalone SAK-CIC310-OSMX2HT device can be easily connected to an external host device via a asynchronous/synchronous serial interface (ASC). This mode is selected by the pin `MODE0=1` at the rising edge of the `RESET` signal. The on-chip SSC can be connected to an external ASC (Synchronous Master Mode) according to [Figure 4-8](#). The configuration of the ASC Host Communication Interface is e.g. for the ASC0 of Infineons TC1796:

Configuration of the ASC0 Port Pins used for MRST, SCLK, and MTSR:

```
// load control register:
// P5.0 is used as ASC0 input signal (ASC0_RX) (MRST).
P5_IOCR0 = (P5_IOCR0 & ~0x000000F0) | 0x00000000;
//
// Alternatively: load control register:
// P5.0 is used as ASC0 output signal (ASC0_RX) (MTSR);
// the push/pull function is activated;
// output driver characteristic: strong driver, sharp edge).
P5_IOCR0 = (P5_IOCR0 & ~0x000000F0) | 0x00000090;
//
// load control register:
```

Host Communication Channels

```
// P6.8 is used as ASC0 output signal (ASC0_RX) (MTRSR);
// the push/pull function is activated;
// output driver characteristic: strong driver, sharp edge.
P6_IOCR8 = (P6_IOCR8 & ~0x000000F0) | 0x000000A0;
//
// Alternatively:
// load control register:
// P6.8 is used as ASC0 input signal (ASC0_RX) (MRST);
P6_IOCR8 = (P6_IOCR8 & ~0x000000F0) | 0x00000000;
// ASC0 receiver input RXD0A (P5.0) selected.
ASC0_PISEL=0;
//
// Alternatively:
// ASC0 receiver input RXD0B (P6.8) selected).
ASC0_PISEL=1;
//
// load control register:
// P5.1 is used as ASC0 output signal (ASC0_TX);
// the push/pull function is activated;
// output driver characteristic: strong driver, sharp edge.
P5_IOCR0 = (P5_IOCR0 & ~0x0000F000) | 0x00009000;
```

Configuration of the Module Clock:

```
// load clock control register:
// enable the ASC0 module;
// clock divider for normal operation mode:
// System clock / 1 (= 75,000 MHz; 13,333 ns).
ASC0_CLC = 0x00000100;
```

Configuration of the ASC0 Baud Rate Generator (reduce serial clock to 2, required baud rate = 9.375 Mbaud at $f_{sys}=75\text{MHz}$)

```
// load ASC0 baud rate time reload register
ASC0_BG = 0x00000000;
```

Configuration of the ASC0 Operation Mode:

```
// load ASC0 control register: 8-bit data synchronous operation;
// receiver is disabled
ASC0_CON = 0x00000000;
```

The RDY output signals the availability of the SSC Slave. This signal can be used to detect overload scenario of the SSC slave device. Baud rate failure can not be check for in this mode, because the ASC module inserts a idle clock cycle every 8 bit causing a baud rate failure for the SSC running in 16 bit modus.

Note: On the ASC in synchronous mode either a transmission or a reception takes place at the same time, regardless whether valid data has been transmitted or received

Host Communication Channels

on both lines (one of the two is disregarded). Therefore application software has to take care that the SSC is switched correctly to either transmission or reception. Even so the SSC transmits and receives always at the same time, the specific SSC protocol of SAK-CIC310-OSMX2HT only has either a valid transmitted data or a valid received data at the same time.

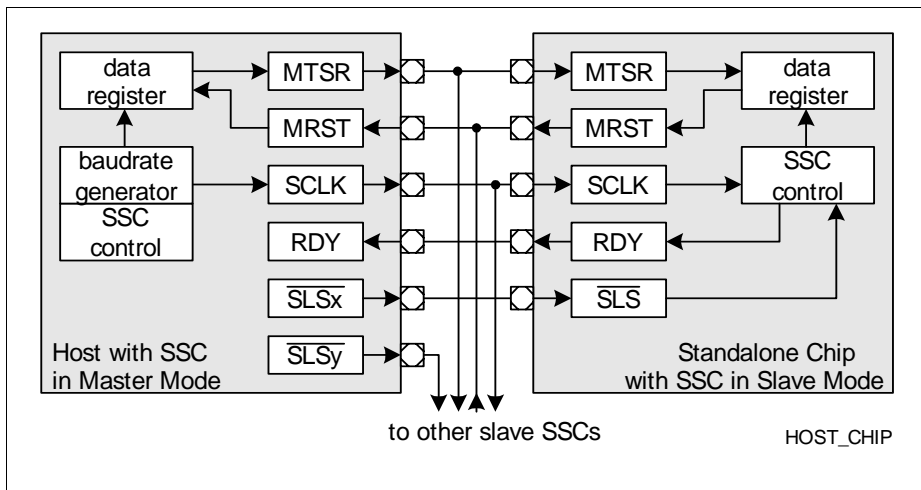


Figure 4-9 SSC in Slave Mode

The Slave Mode communication protocol is optimized for a transfer of data halfwords (16 bit halfwords) located at consecutive addresses. Upon the transmission of a single address halfword by the external SSC, a data stream is returned or expected by the SAK-CIC310-OSMX2HT's SSC as long as the $\overline{\text{SLS}}$ pin is held on 0 level (active state) and the SCLK pin is provided with a clock signal. The signal $\overline{\text{SLS}}$ has to be set to 1 in order to finalize a single halfword or block access. An access to a different register location has to be indicated by a falling edge of the $\overline{\text{SLS}}$ signal, initializing a new access.

The external SSC master has to transmit a transaction header after activating the $\overline{\text{SLS}}$ signal. This transaction header contains the CMD bit, the INCE bit, and a 14 bit address bit field (see [Figure 4-10](#)). CMD=0 indicates a read and CMD=1 indicates a write requested by the SSC master. If the control bit INCE is set to 1, the contents of the address register is automatically incremented by 2 after each transfer of a halfword. The automatic increment of the address register is stopped at 0xFFFF in order to avoid an unintended overwrite of the following data space. When INCE is held on 0, the memory location defined by the transmitted address halfword, is continuously reread or rewritten.

Host Communication Channels

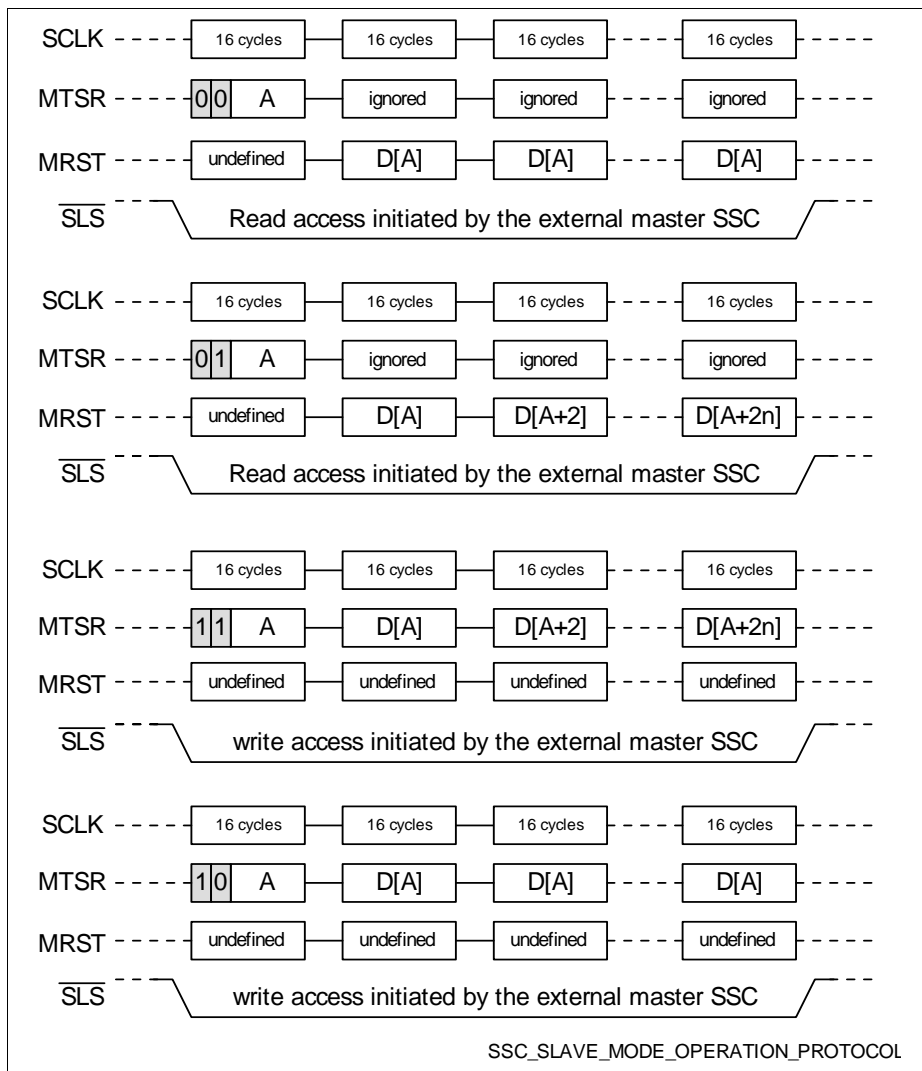


Figure 4-10 Slave Mode Communication Protocol

In **Figure 4-10**, the first transferred halfword of the master contains the 14-bit address (A), the following halfwords contain the desired data (D) read/written at the selected address. In this example, bit INCE has been set to 1 in order to allow an access to

Host Communication Channels

consecutive addresses (A to A+2n) and set to 0 to illustrate an access to unvaried address (A).

4.5.1 Single Read Access via SSC

Table 4-8 and **Figure 4-11** illustrate the actions on host side (SSC master) and on SAK-CIC310-OSMX2HT side (SSC slave) for a single halfword read operation via the SSC communication channel.

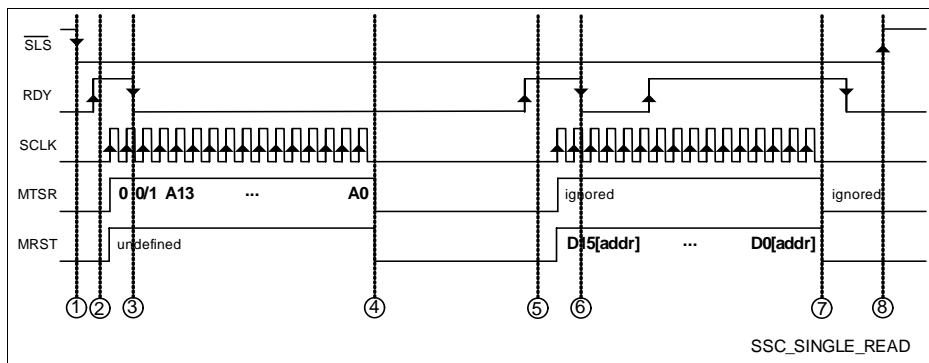


Figure 4-11 Single SSC Read Access

The following table describes the functionality on the SSC host side and on the SSC slave side for a single read access to the Standalone SAK-CIC310-OSMX2HT device. The 14 bit address given by the host is named "A".

Table 4-8 Single Read Access

Action	Host action	Device action
1	Activation of the $\overline{\text{SLS}}$ output signal to indicate the start of a new communication sequence.	
1		As soon as the falling edge of the $\overline{\text{SLS}}$ input signal has been detected, output RDY becomes active (1), indicating that the Standalone SAK-CIC310-OSMX2HT device is ready for data exchange.

Host Communication Channels

Table 4-8 Single Read Access (cont'd)

Action	Host action	Device action
2	The desired address (A), the read-indication bit (0) and the INC bit ("X") have to be prepared and the transmission of the first halfword is started as soon as RDY=1 has been detected.	-
3	-	The detection of the first clock edge for the current halfword transfer resets output RDY.
4	Reception of a halfword with an undefined value.	After complete reception of the transferred halfword, the read-indication, the INC bit and the 14 address bits (A) are stored. Output RDY is set to indicate that the required data halfword has been read from the selected address (D[A]) and can be transferred.
5	The transmission of the next halfword (value without impact) is started as soon as RDY=1 has been detected.	-
6	-	The detection of the first clock edge for the current halfword transfer resets output RDY.
7	The desired data halfword (D[A]) has been received.	
8	Deactivation of the $\overline{\text{SLS}}$ output signal to indicate the end of the communication sequence.	As soon as input $\overline{\text{SLS}}$ =1 has been detected, output RDY is reset. If this is detected before having set RDY, RDY will not be set (see action 7). The communication sequence is finished.

4.5.2 Consecutive Read Accesses via SSC

Figure 4-12 illustrates the actions on the host side (master) and on the SAK-CIC310-OSMX2HT side (SSC slave) for a consecutive halfword read operation via the SSC communication channel without automatic address increment (INCE=0).

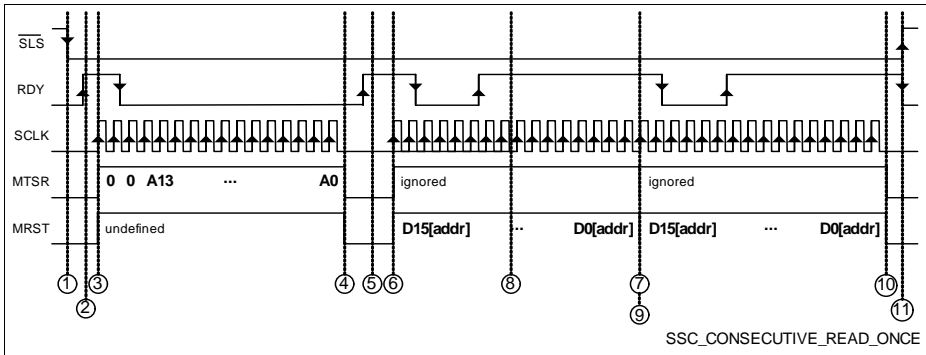


Figure 4-12 Consecutive SSC Read Accesses (INCE=0)

Figure 4-13 illustrates the actions on the host side (SSC master) and on the SAK-CIC310-OSMX2HT side (SSC slave) for a consecutive halfword read operation via the SSC communication channel with automatic address increment (INCE=1).

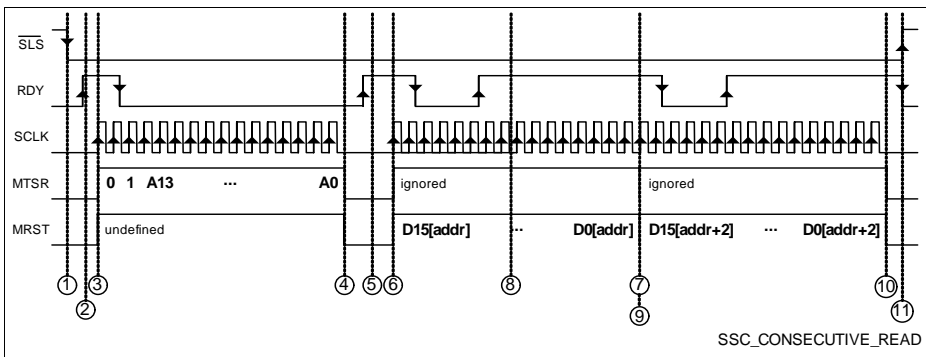


Figure 4-13 Consecutive SSC Read Accesses (INCE=1)

Table 4-9 describes the functionality on the host side and on the slave side for consecutive read accesses to the Standalone SAK-CIC310-OSMX2HT device with (INCE=1) and without (INCE=0) an automatic address increment. The 14 bit address given by the host is named “A” (for INCE=1 to indicate that this address is increment by the slave) or “A0” (for INCE=0 to indicate that this address is constant).

Table 4-9 Consecutive Read Accesses

Action	Host action	Device action
1	Activation of the $\overline{\text{SLS}}$ output signal to indicate the start of a new communication sequence.	As soon as the falling edge of the $\overline{\text{SLS}}$ input signal has been detected, output RDY becomes active (1), indicating that the Standalone SAK-CIC310-OSMX2HT device is ready for data exchange.
2	<p>INCE=0: The desired address (A0), the read-indication bit (0) and the INC bit (0) have to be prepared and the transmission of the first halfword is started as soon as RDY=1 has been detected. The read-indication and the INC bit is valid for the complete communication sequence.</p> <p>INCE=1: The desired address (A) and the read-indication bit (0) and the INC bit (1) have to be prepared and the transmission of the first halfword is started as soon as RDY=1 has been detected. The read-indication and the INC bit is valid for the complete communication sequence.</p>	-
3	-	The detection of the first clock edge for the current halfword transfer resets output RDY.

Host Communication Channels

Table 4-9 Consecutive Read Accesses (cont'd)

Action	Host action	Device action
4	Reception of a halfword with an undefined value.	<p>INCE=0: After complete reception of the transferred halfword, the read-indication, the INC bit and the 14 address bits (A0) are stored. Output RDY is set to indicate that the required data halfword has been read from the selected address (D[A0]) and can be transferred.</p> <p>INCE=1: After complete reception of the transferred halfword, the read-indication, the INC bit and the 14 address bits (A) are stored. Output RDY is set to indicate that the required data halfword has been read from the selected address (D[A]) and can be transferred.</p>
5	The transmission of the next halfword (value without impact) is started as soon as RDY=1 has been detected.	-
6	-	The detection of the first clock edge for the current halfword transfer resets output RDY.

Host Communication Channels

Table 4-9 Consecutive Read Accesses (cont'd)

Action	Host action	Device action
7	<p>INCE=0: The desired data halfword (D[A0]) has been received.</p> <p>INCE=1: The desired data halfword (D[A]) has been received.</p>	<p>INCE=0: After complete reception of the transferred halfword, output RDY is set to indicate that the required data halfword D[A0] is available and can be transferred.</p> <p>INCE=1: The formerly used address is increment by 2 in order to generate the new one. After complete reception of the transferred halfword, output RDY is set to indicate that the required data halfword has been read from the new selected address (D[A+2]) and can be transferred.</p>
8	The transmission of the next halfword (value without impact) is started as soon as RDY=1 has been detected.	-
9	-	The detection of the first clock edge for the current halfword transfer resets output RDY.

Host Communication Channels

Table 4-9 Consecutive Read Accesses (cont'd)

Action	Host action	Device action
10	<p>INCE=0: The desired data halfword (D[A0]) has again been received.</p> <p>INCE=1: The desired data halfword (D[A+2n]) has been received.</p>	<p>INCE=0: After complete reception of the transferred halfword, output RDY is set to indicate that the required data halfword D[A0] is available and can be transferred.</p> <p>INCE=1: The formerly used address is increment by 2 in order to generate the new one. After complete reception of the transferred halfword, output RDY is set to indicate that the required data halfword has been read from the new selected address (D[A+2n+2]) and can be transferred.</p>
11	Deactivation of the $\overline{\text{SLS}}$ output signal to indicate the end of the communication sequence.	<p>As soon as input $\overline{\text{SLS}}=1$ has been detected, output RDY is reset. If this is detected before having set RDY, RDY will not be set (see action 7). The communication sequence is finished.</p>

4.5.3 Single Write Access via SSC

Figure 4-14 and **Table 4-10** illustrate the actions on the host side (SSC master) and on the Standalone SAK-CIC310-OSMX2HT side (SSC slave) for a single halfword write operation via the SSC communication channel.

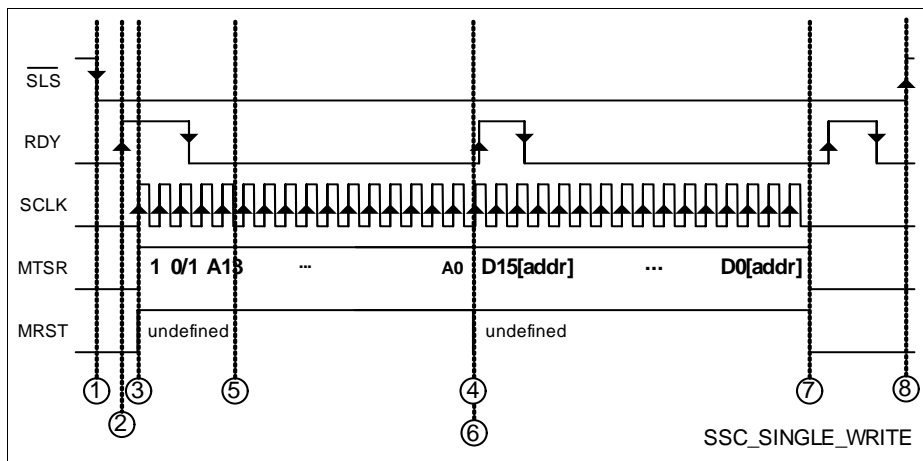


Figure 4-14 Single SSC Write Access

Table 4-10 Single Write Access

Action	Host action	Device action
1	Activation of the $\overline{\text{SLS}}$ output signal to indicate the start of a new communication sequence.	As soon as the falling edge of the $\overline{\text{SLS}}$ input signal has been detected, output RDY becomes active (1), indicating that the Standalone SAK-CIC310-OSMX2HT device is ready for data exchange.
2	The desired address (A), the write-indication bit (1) and the INC bit ("X") have to be prepared and the transmission of the first halfword is started as soon as RDY=1 has been detected.	-
3	-	The detection of the first clock edge for the current halfword transfer resets output RDY.

Host Communication Channels

Table 4-10 Single Write Access (cont'd)

Action	Host action	Device action
4	Reception of a halfword with an undefined value.	After complete reception of the transferred halfword, the write-indication, the INC bit and the 14 address bits (A) are stored. Output RDY is set to indicate that the next halfword can be transferred.
5	The transmission of the data halfword to be written is started as soon as RDY=1 has been detected.	-
6	-	The detection of the first clock edge for the current halfword transfer resets output RDY.
7	The transmission of the data halfword is finished.	After complete reception of the transferred halfword, output RDY is set to indicate that the data halfword has been written to the selected address (D[A]).
8	Deactivation of the $\overline{\text{SLS}}$ output signal to indicate the end of the communication sequence.	As soon as input $\overline{\text{SLS}}=1$ has been detected, output RDY is reset. If this is detected before having set RDY, RDY will not be set (see action 7). The communication sequence is finished.

4.5.4 Consecutive Write Access via SSC

Figure 4-15 illustrates the actions on the host side (SSC master) and on the SAK-CIC310-OSMX2HT side (SSC slave) for a consecutive halfword write operation via the SSC communication channel without automatic address increment (INCE=0).

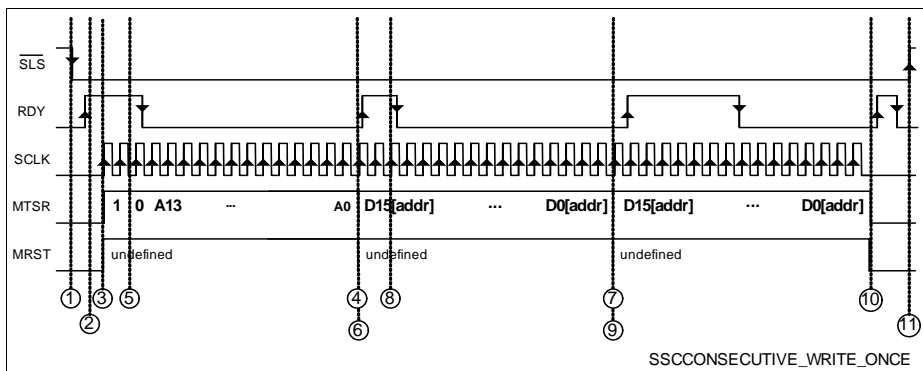


Figure 4-15 Consecutive SSC Write Accesses (INCE=0)

Figure 4-16 illustrates the actions on the host side (master) and on the SAK-CIC310-OSMX2HT side (slave) for a consecutive halfword write operation via the SSC communication channel with automatic address increment (INCE=1).

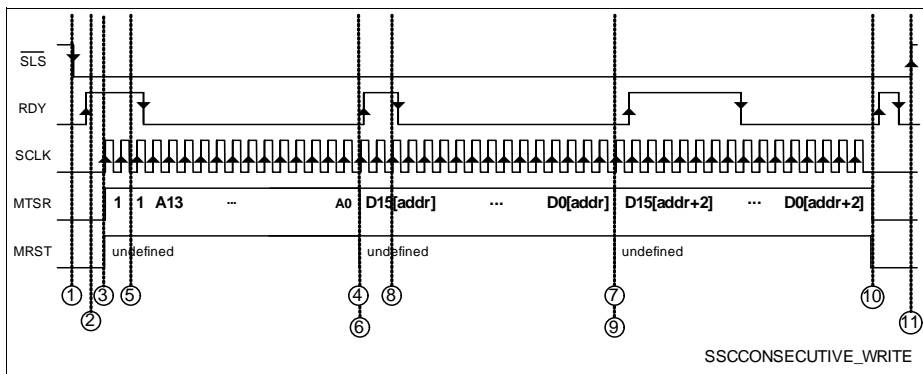


Figure 4-16 Consecutive SSC Write Accesses (INCE=1)

Table 4-11 describes the functionality on the host side and on the slave side for consecutive write accesses to the Standalone SAK-CIC310-OSMX2HT device with (INCE=1) and without (INCE=0) an automatic address increment. The 14 bit address given by the host is named “A” to indicate that this address is only transmitted once.

Table 4-11 Consecutive Write Accesses

Action	Host action	Device action
1	Activation of the $\overline{\text{SLS}}$ output signal to indicate the start of a new communication sequence.	As soon as the falling edge of the $\overline{\text{SLS}}$ input signal has been detected, output RDY becomes active (1), indicating that the Standalone SAK-CIC310-OSMX2HT device is ready for data exchange.
2	<p>INCE=0: The desired address (A), the write-indication bit (0) and the INC bit (0) have to be prepared and the transmission of the first halfword is started as soon as RDY=1 has been detected. The write-indication and the INC bit is valid for the complete communication sequence.</p> <p>INCE=1: The desired address (A) and the write-indication bit (0) and the INC bit (1) have to be prepared and the transmission of the first halfword is started as soon as RDY=1 has been detected. The write-indication and the INC bit is valid for the complete communication sequence.</p>	-
3	-	The detection of the first clock edge for the current halfword transfer resets output RDY.
4	Reception of a halfword with an undefined value.	After complete reception of the transferred halfword, the write-indication, the INC bit and the 14 address bits (A) are stored. Output RDY is set to indicate that the next halfword can be transferred.

Host Communication Channels

Table 4-11 Consecutive Write Accesses (cont'd)

Action	Host action	Device action
5	The transmission of the first data halfword to be written is started as soon as RDY=1 has been detected.	-
6	-	The detection of the first clock edge for the current halfword transfer resets output RDY.
7	The transmission of the first data halfword is finished.	<p>INCE=0: After complete reception of the transferred halfword, output RDY is set to indicate that the data halfword has been written to the selected address (D[A]) and a new data halfword can be transferred. The value of the selected target address is not modified.</p> <p>INCE=1: After complete reception of the transferred halfword, output RDY is set to indicate that the data halfword has been written to the selected address (D[A]) and a new data halfword can be transferred. The value of the selected target address is incremented by 2 (as long as the value of 0xFFFF is not exceeded).</p>
8	The transmission of the second data halfword to be written is started as soon as RDY=1 has been detected.	-
9	-	The detection of the first clock edge for the current halfword transfer resets output RDY.

Host Communication Channels

Table 4-11 Consecutive Write Accesses (cont'd)

Action	Host action	Device action
10	The transmission of the n^{th} data halfword is finished.	<p>INCE=0: After complete reception of the transferred halfword, output RDY is set to indicate that the data halfword has been written to the selected address (D[A]) and a new data halfword can be transferred. The value of the selected target address is not modified.</p> <p>INCE=1: After complete reception of the transferred halfword, output RDY is set to indicate that the data halfword has been written to the selected address (D[A + 2n - 2]) and a new data halfword can be transferred. The value of the selected target address is increment by two (as long as the value of 0xFFFF is not exceeded).</p>
11	Deactivation of the $\overline{\text{SLS}}$ output signal to indicate the end of the communication sequence.	<p>As soon as input $\overline{\text{SLS}}=1$ has been detected, output RDY is reset. If this is detected before having set RDY, RDY will not be set (see action 7).</p> <p>The communication sequence is finished.</p>

4.5.5 Error Handling

If an error condition (e.g. by the transmission of a lower or a higher number than 16 bits or by a spike on the SCLK line) is detected by the baud rate error detection, the RDY signal will become inactive. Furthermore, a received baud rate exceeding the limits of double or half of the selected baud rate will lead to the same error (has to be deselected if communicating with an ASC as Host interface, because the ASC will generate a clock idle time after every 8 bit data. The error condition can be checked by the host, e.g. by a time-out function. The baud rate error detection has to be explicitly enabled by program.

Host Communication Channels

If an error has been detected, the host must deactivate the $\overline{\text{SLS}}$ signal to finish the communication cycle. The host has to wait for at least 5 clock cycles after the rising edge of $\overline{\text{SLS}}$ before a new communication cycle can be started by activating the $\overline{\text{SLS}}$ signal. The end of the internal SSC error recovery process is indicated by the RDY line becoming active in the next communication cycle, which indicates that the transmission of a new halfword can be started. After the recovery process the past error situation can be checked by program by reading of the corresponding status register.

5 Direct Memory Access Controller (DMA)

This chapter describes the Direct Memory Access Controller (DMA) of the SAK-CIC310-OSMX2HT. This chapter contains the following sections:

- Functional description of the DMA kernel (see [Section 5.1](#)).
- Kernel register description (see [Section 5.2](#)).
- SAK-CIC310-OSMX2HT implementation specific details (see [Section 5.3](#)).

The DMA executes DMA transactions from a source address location to a destination address location, without intervention of the host CPU. One DMA transaction is controlled by one DMA channel. Each DMA channel has assigned its own channel register set. The total of 8 channels is provided by the DMA.

5.1 DMA Controller Description

5.1.1 Features

- 8 independent DMA channels
 - 8 selectable request inputs per DMA channel
 - Programmable priority of DMA channels within the DMA sub-block (2 levels)
 - Software and hardware DMA request generation
 - Hardware requests by selected peripherals
- Individually programmable operation modes for each DMA channel
 - Single Mode: stops and disables DMA channel after a predefined number of DMA transfers
 - Continuous Mode: DMA channel remains enabled after a predefined number of DMA transfers; DMA transaction can be repeated.
 - Programmable address modification
- Full 32-bit addressing capability of each DMA channel
 - 4 Gbyte address range
 - Support of circular buffer addressing mode
- Programmable data width of a DMA transaction: 8-bit, 16-bit, or 32-bit
- Individual register set for each DMA channel
 - Source and destination address register
 - Channel control and status register
 - Transfer count register
- Flexible interrupt generation

Direct Memory Access Controller (DMA)

5.1.2 Definition of Terms

DMA Transaction

A DMA transaction is composed of several (at least one) DMA transfers. The Transfer Count defines the number of DMA transfers within one DMA transaction.

DMA Transfer

A DMA transfer is composed of 1, 2, 4, 8, or 16 moves. For hardware triggered DMA operations this action is done per request.

DMA Move

A DMA move is an operation which consists always of two parts:

- A source move that loads data from a data source into the DMA controller
- A destination move that puts data from the DMA controller to a data destination

Within a DMA move data is always moved from the data source via the DMA controller to the data destination. The data width of source move and destination move are always identical (8-bit, 16-bit, or 32-bit).

Example:

1024 words transaction can be composed of 256 transfers of 4 moves or 128 transfers of 8 moves.

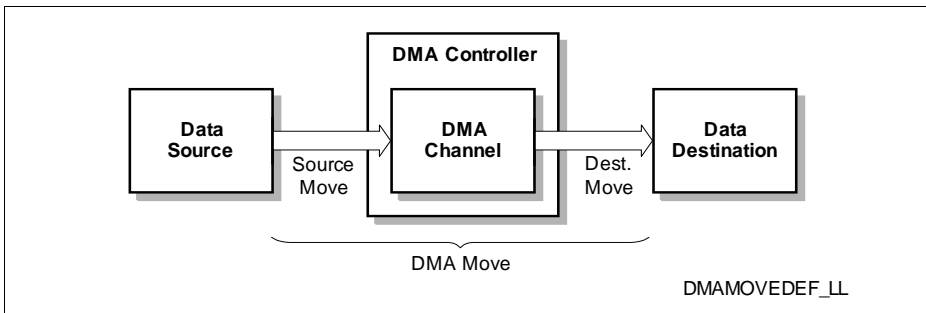


Figure 5-1 DMA Terms Definitions

Direct Memory Access Controller (DMA)

5.1.3 DMA Principle

The DMA controller supports DMA moves from one address location to another one. DMA moves can be requested either by hardware or by software. DMA hardware requests are triggered by specific request lines from the peripheral modules (see [Figure 5-2](#)).

The DMA controller consists of a control unit and one DMA sub-block. Once configured, the sub-block of the DMA controller is able to act as a master on the bus.

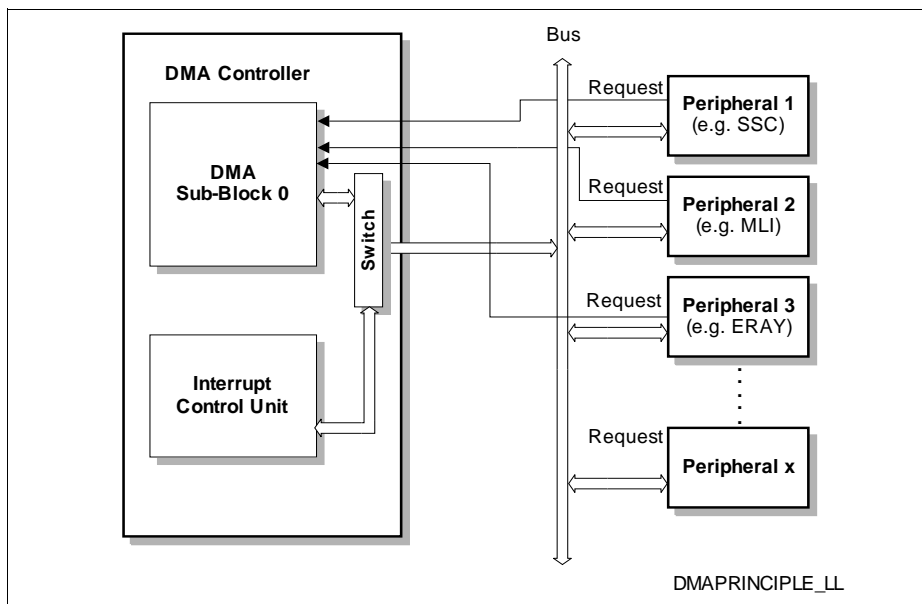


Figure 5-2 DMA Principle

Direct Memory Access Controller (DMA)

5.1.4 DMA Block Diagram

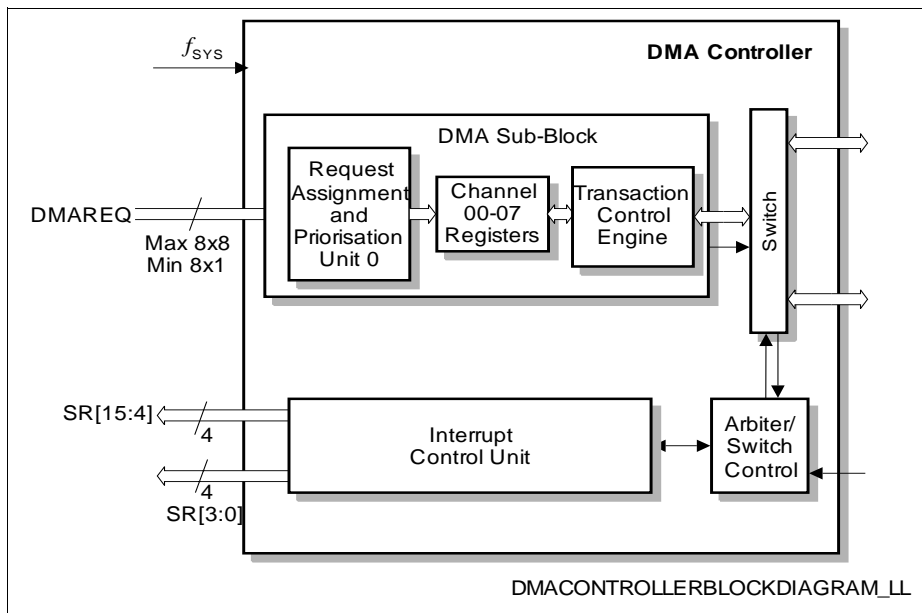


Figure 5-3 DMA Block Diagram

5.1.5 DMA Operation Functionality

Each DMA channel has its own register set. The actual transfer count information during an active DMA transaction can be read in register **CHSR0n** ($n = 0-7$) ($n = 0-7$, for channel number) as status information.

A DMA transaction is initiated either by software (immediately started after the channel activation) or by hardware via the DMA request input **ch0n_req**. After completion of a DMA transaction, a service request signal can be generated.

Direct Memory Access Controller (DMA)

5.1.5.1 Shadow Registers

There are two shadowed registers in the DMA for each channel: a shadowed address, and a shadowed counter register.

The shadowed address register is used by the software when the DMA channel is busy. The shadow registers have the same behavior for both Single and Continuous Mode. (See [Figure 5-4](#) and [Figure 5-5](#))

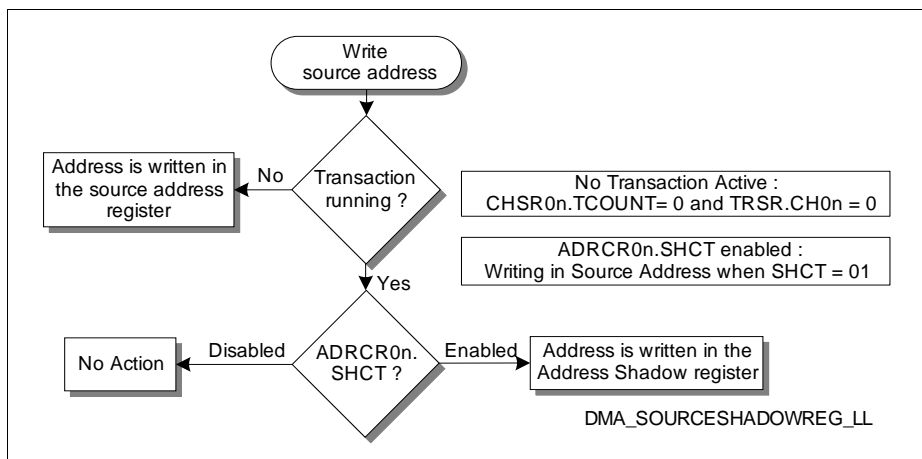


Figure 5-4 Source Shadow Register

Example: While a data source, e.g. the E-Ray module (source address is fixed) delivers data words that are written to a buffer in memory, the shadow mechanism can be used to program the location of a new buffer address for the next transaction. After the end of the current transaction, the new transaction can start at a new address without host intervention.

Direct Memory Access Controller (DMA)

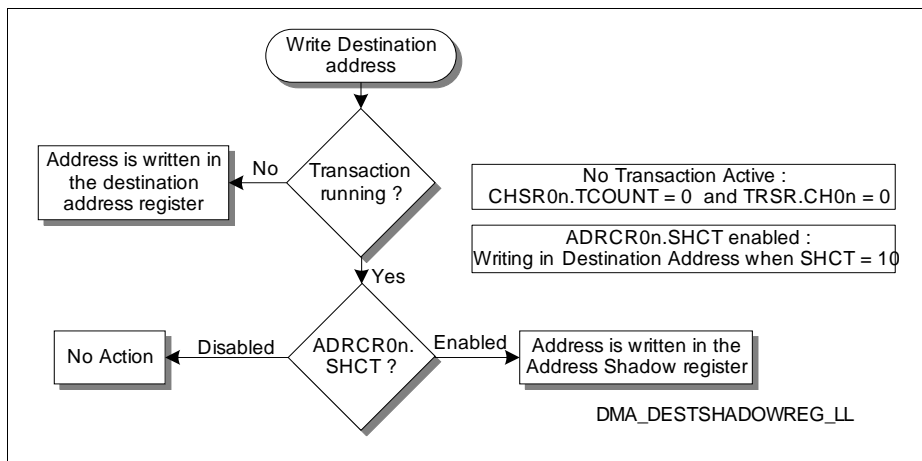


Figure 5-5 Destination Shadow Register

At the start of a new transaction, the value of the shadow address register **SHADR0n (n = 0-7)** is transferred to the source or destination address register and the shadow register is cleared (if enabled by **ADRCR0n (n = 0-7).SHCT** and the shadow contents is valid).

The shadowed counter register **CHCR0n (n = 0-7).TREL** of channel 0n can also be programmed even if the channel is processing a transaction. The value will be transferred in **CHSR0n (n = 0-7).TCOUNT** when a new transaction is started.

No reload of address or counter will be done if **CHSR0n.TCOUNT** is not equal to 0.

Note: The reprogramming of channel specific values (except for the selected address shadow register) should be avoided while a channel is active.

Direct Memory Access Controller (DMA)

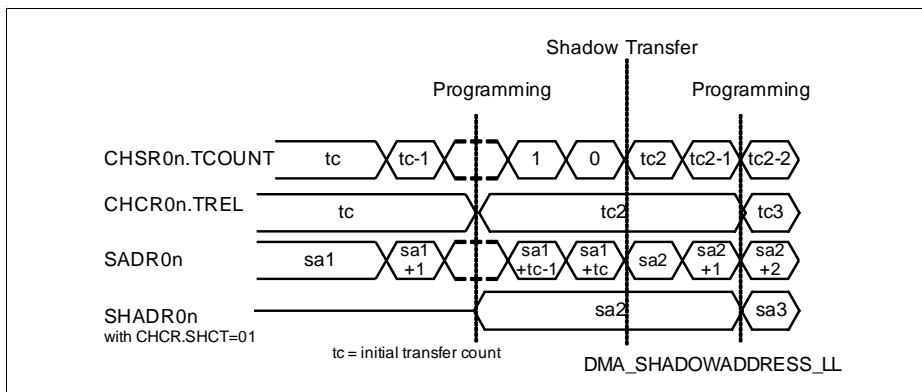


Figure 5-6 Shadow Address and Counter Example

If the transaction of DMA channel 0n is stopped (**TRSR.HTRE0n** (**n = 0-7**) = 0 or **TRSR.CH0n** (**n = 0-7**) = 0), the transaction will finish with the values written before. After a new activation, the transaction will continue at the point it stopped. The new value for the address and the reload value for CHSR0n.TCOUNT0n are only taken into account when a new transaction is started. They are kept consistent during a running transaction.

Direct Memory Access Controller (DMA)

5.1.5.2 DMA Channel Request Control

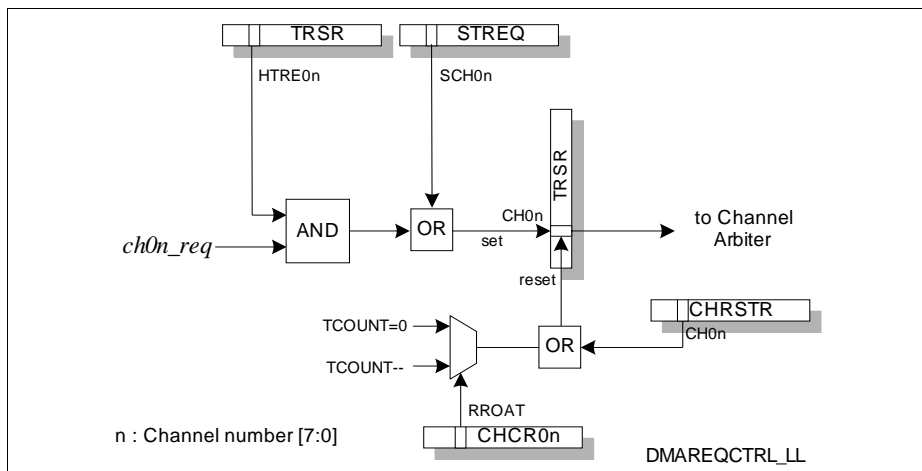


Figure 5-7 Channel Request Control

5.1.5.3 DMA Channel Operation Mode

The operation mode is individually programmable for each DMA channel $0n$. A channel can operate either in one of two modes:

- Single Mode (independent start for each transactions)
- Continuous Mode (several transactions back-to-back possible)

In Single Mode, the DMA channel $0n$ is disabled (**TRSR.HTRE0n** = 0 and **TRSR.CH0n** = 0) after the last DMA transfer of a DMA transaction. For the start of the next DMA transaction, DMA channel $0n$ must be set again (**TRSR.HTRE0n** = 1 or **TRSR.CH0n** = 1).

In Continuous Mode, the DMA channel $0n$ remains active for HW requests (**TRSR.HTRE0n** = 1) when a transaction has been finished.

If **TRSR.HTRE0n** is cleared (with **HTREQ.DCH0n** ($n = 0-7$) set), while doing a transfer, the current transfer it will be finished before stopping the DMA transaction, for counter and address consistency.

Bit **TRSR.HTRE0n** can be set and cleared by SW (writing to register **HTREQ**) and it can be cleared by HW at the end of a transaction in Single Mode (depending on **CHCR0n** ($n = 0-7$).**CHMODE**).

*Note: The bit **TRSR.CH0n** is set automatically each time a rising edge is detected at the selected $ch0n_req$ input line while **TRSR.HTRE0n** is set.*

Direct Memory Access Controller (DMA)

Software Controlled Single Mode (one trigger per transaction)

This mode is selected by $\text{TRSR.HTRE0n} = 0$ and $\text{CHCR0n (n = 0-7).RROAT} = 1$.

In Software Controlled Single Mode, setting request bit $\text{STREQ.SCH0n (n = 0-7)}$ causes the DMA transaction of channel 0n to be started. The DMA transaction consists of a predefined number of DMA transfers (transfer count or tc), as defined in CHSR0n.TCOUNT . After each transfer CHSR0n.TCOUNT is decremented. When the counter reaches $\text{CHICR0n (n = 0-7).IRDV}$, or when a transfer is finished, the interrupt signal sr0n can be activated (see also CHICR0n.INTCT). When the counter reaches the value 0, the DMA channel 0n becomes disabled ($\text{TRSR.CH0n} = 0$) and the interrupt signal sr can be activated. Setting STREQ.SCH0n again starts the next DMA transaction with the parameters as defined in the channel register set. A running DMA transaction is indicated by a set channel active flag TRSR.CH0n . While $\text{TRSR.HTRE0n} = 0$, HW DMA requests are not taken into account.

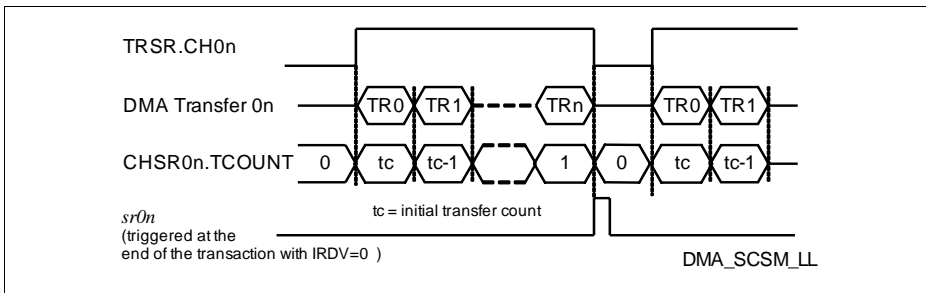


Figure 5-8 Software Controlled Single Mode Operation

Hardware Controlled Single Mode

This mode is selected by $\text{CHCR0n.CHMODE} = 0$ and $\text{CHCR0n.RROAT} = 0$.

In this Hardware Controlled Single Mode, setting HTREQ.ECH0n causes the DMA transaction to be activated. The DMA transaction consists of a predefined number of DMA transfers (tc), as defined in CHSR0n.TCOUNT . A DMA transfer of the DMA transaction is executed whenever the DMA request input line of DMA channel 0n becomes active. After each transfer TCOUNT is decremented. When CHSR0n.TCOUNT reaches CHICR0n.IRDV , or when a transfer finishes, the interrupt signal sr can be activated. When CHSR0n.TCOUNT reaches the value 0, the DMA channel 0n becomes disabled ($\text{TRSR.HTRE0n} = 0$). Setting HTREQ.ECH0n again starts the next DMA transaction with the parameters defined in the channel register set. The running DMA transaction is indicated by TRSR.HTRE0n set.

If $\text{CHCR0n.CHMODE} = 0$, bit TRSR.HTRE0n is automatically cleared when the end of the transaction is reached ($\text{TCOUNT} = 0$) → Single Mode.

In order to start the next transaction, the SW has to set TRSR.HTRE0n again.

Direct Memory Access Controller (DMA)

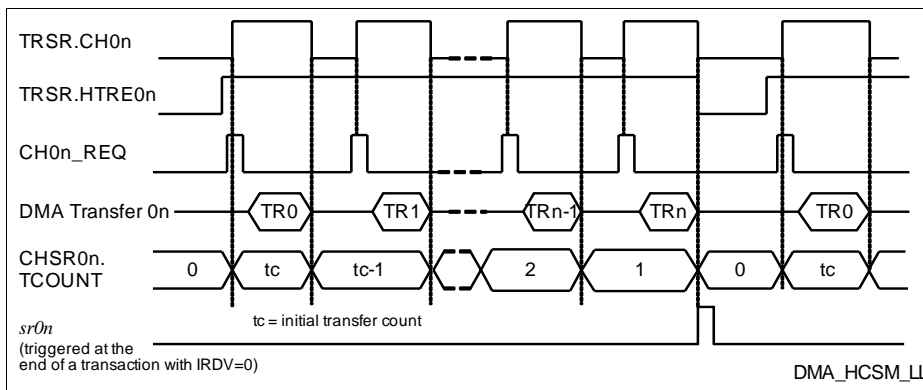


Figure 5-9 Hardware Controlled Single Mode Operation

When bit **CHCR0n.RROAT** = 0, bit **TRSR.CH0n** will be cleared after each transfer. So the software can initiate a transaction by writing **STREQ.SCH0n**, and the transaction will be stopped after the first transfer. While the hardware requests are enabled (**TRSR.HTRE0n** = 1), the transaction can continue with hardware control. (see [Figure 5-10](#)).

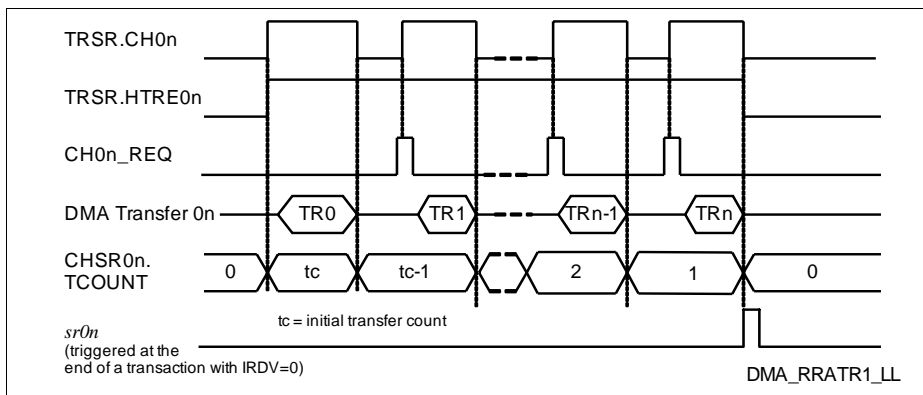


Figure 5-10 Transaction Start by SW, Continuation by HW

Hardware Controlled Continuous Mode

This mode is selected by **CHCR0n (n = 0-7).CHMODE** = 1 and **CHCR0n (n = 0-7).RROAT** = 0.

Direct Memory Access Controller (DMA)

In Hardware Controlled Continuous Mode, setting of **HTREQ.ECH0n (n = 0-7)** causes the DMA transaction to be activated. The DMA transaction consists of a predefined number of DMA transfers (tc), as defined in CHSR0n.TCOUNT. A DMA transfer of the DMA transaction is executed whenever the DMA request input line of DMA channel 0n becomes active. After each transfer TCOUNT is decremented. Each time CHSR0n.TCOUNT reaches CHICR0n.IRDV, or when a transfer finishes, the interrupt signal sr0n of DMA channel 0n can be activated (see **“Interrupt Signal Generation” on Page 5-14**). When CHSR0n.TCOUNT reaches 0000_H, the DMA channel starts a new DMA transaction with the parameters defined in the channel register set. When bit CHCR0n.CHMODE is cleared, the current DMA transaction of DMA channel 0n will finish when CHSR0n.TCOUNT = 0000_H. During Continuous Mode (CHCR0n.CHMODE = 1) TRSR.HTREQ0n is not automatically cleared.

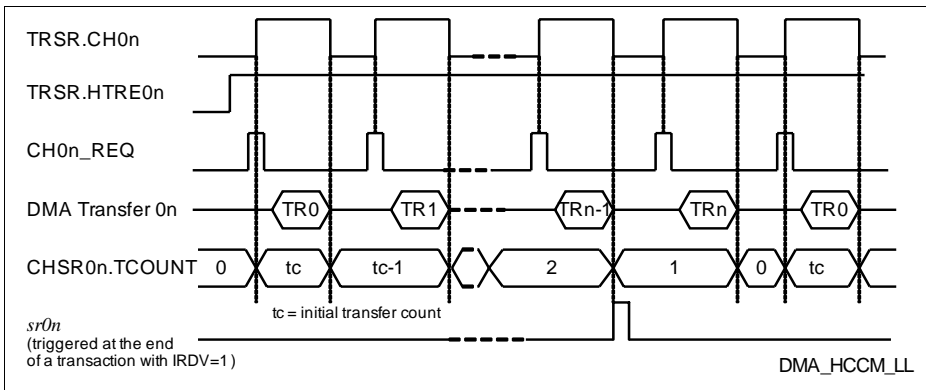


Figure 5-11 Hardware Controlled Continuous Mode Operation

Continuous Mode With Request Reset After Complete Transaction

If bit CHCR0n.RROAT = 1, the requests **TRSR.CH0n (n = 0-7)** are cleared only after the complete transaction. The requests will not be cleared after each transfer, so any request will trigger a complete transaction. (see **Figure 5-12**). In this Continuous Mode, the HW can request one complete transaction after the other, because the bit TRSR.HTREQ0n is not automatically cleared after a transaction (CHCR0n.CHMODE = 1).

Direct Memory Access Controller (DMA)

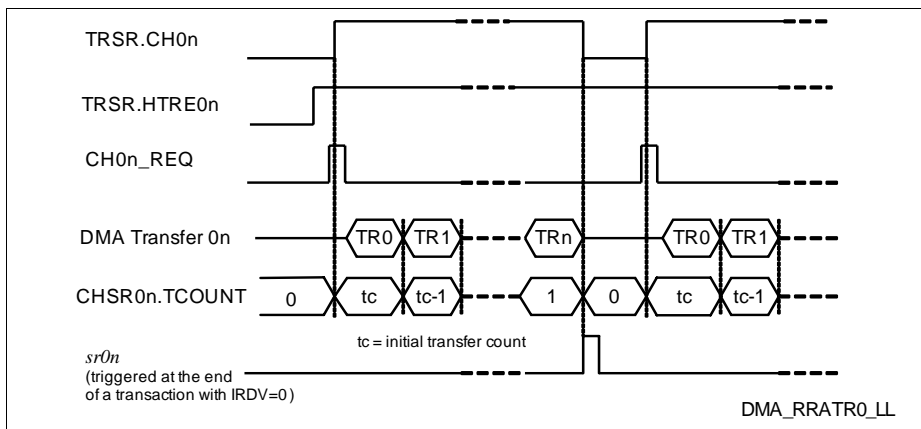


Figure 5-12 One Trigger, One Transaction

5.1.5.4 Move Count

It defines the number of moves (consisting of one read and one write each) to be done in each transfer. It allows the user to indicate to the DMA the number of moves to be done after one request. The number of moves per transfer is selected by the block mode settings (**CHCR0n (n = 0-7).BLKM**).

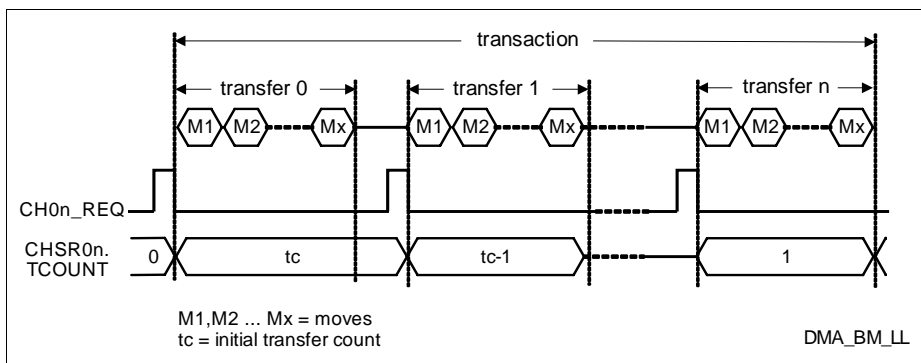


Figure 5-13 Transfer Mode

5.1.5.5 Request Lost

If a software or a hardware request is detected for channel 0n while TRSR.CH0n is set, a request lost event occurs. This error event indicates that the DMA is already

Direct Memory Access Controller (DMA)

processing a transfer, and that an another transfer is requested before the end of the first one. In this case, bit **ERRSR.TRL0n (n = 0-7)** will be set and an interrupt can be generated.

5.1.5.6 Circular Buffer

In order to support wrap around, the software can read the **SADR0n (n = 0-7)** or **DADR0n (n = 0-7)** registers with a new address in reaction to the interrupt indicating that tc transfers have been done. In addition to this function, the bit fields **ADRCR0n (n = 0-7).CBLS** and **ADRCR0n (n = 0-7).CBLD** can be used to program an automatic wrap around.

These register bit fields defines the bit position used to detect the end of the buffer, only buffer length of 2^n can be selected.

If **ADRCR0n.CBLS** or **ADRCR0n.CBLD** is set to n , after the address increment when the address bit n changes, all the lower address bits ($n - 1$ to 0) do the wrap around (increment or decrement normally as programmed), and all upper bits (31 to n) remain unchanged. In order to access always the same address, all address bits have to be "frozen".

The base address of the circular buffer must always be aligned to a multiple integer value of its size. To use **ADRCR0n.CBLS** or **ADRCR0n.CBLD**, the buffer size of the circular buffer must always be a power of two. In **Figure 5-14**, the light grayed memory space in the left mapping is not aligned to a multiple value of its size (represented in dashed lines). In this example, the buffer with a length of 4 byte must start at the address 04_H or 00_H .

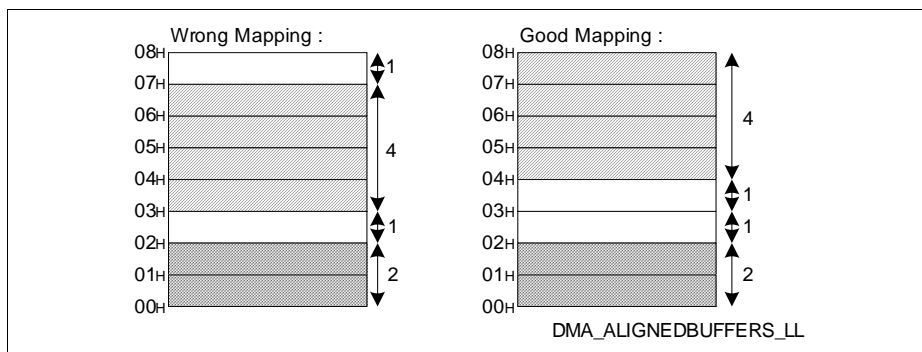


Figure 5-14 Aligned Buffers

5.1.5.7 Interrupt Signal Generation

The DMA has a flexible interrupt signal generation unit. It can generate interrupts signals:

- Every time a transfer is done,
- On a specific value of CHSR0n.TCOUNT
- Or when the value read by the move engine is equal to the move engine pattern.

Depending on the bit field **CHICR0n (n = 0-7).INTCT**, an interrupt signal can generated each time TCOUNT is decremented. It is also possible to generate an interrupt signal when the end of the transaction approaches. Register CHICR0n.IRDV contains the value of the transfer counter that can trigger an interrupt signal. When CHSR0n.TCOUNT = CHICR0n.IRDV an interrupt signal can be generated. It allows the host CPU to react before the end of the transaction is reached if IRDV > 0.(see [Figure 5-11](#))

In these two cases, interrupts signal will be generated after the end of a transfer.

Interrupt signals can also be generated when wrap around of a source or destination address occurs, or when a pattern has been recognized in a move engine read. The wrap around interrupt signal share the interrupt pointer with the pattern detection interrupt.

5.1.5.8 Pattern Detection

The move engine of the DMA can detect a programmable pattern inside the data stream that is handled. Pattern detection can be enabled/disabled independently for each channel. A channel being related exactly to one move engine, the pattern must be written in the move engine registers **ME0PR.PAT00** to **ME0PR.PAT03**. As the patterns are stored in the move engine, the same patterns can be used for all the channels connected to the same move engine.

The bit field **CHCR0n (n = 0-7).PATSEL** configures how the pattern recognition is activated. The pattern recognition checks the data read by the DMA to find a pattern match. The pattern width (8, 16 or 32 bits) to be checked depends on the selected data width.

For 8 bit moves, always LXH is stored in **CHSR0n (n = 0-7).LXO**. For 16 bit moves, the result of the comparison LXH is stored in CHSR0n.LXO if the source address is selected to be decremented in order to be available for comparison in the next move action. If the source address is selected to be increment, LXL is stored in CHSR0n.LXO. Same for 32 bit moves.

Figure 5-15 shows the structure of the pattern recognition part. For the 32 bit comparison, no mask features are available (HH and HL check for equal-to, LH and LL check also for equal-to with a mask of 00_H, matching condition: HH=1 AND HL=1 AND LH=1 AND LL=1). For the 16 bit and 8 bit comparisons, the higher two byte of the pattern register can be used as acceptance (filter) mask.

Direct Memory Access Controller (DMA)

For the 16 bit comparison, the status LL=1 AND LH=1 indicate a pattern match if the pattern is aligned to the read move. If the pattern is not aligned to the read move, the condition LXO=1 AND LXL=1 (for source address decrement) or LXO=1 AND LXH=1 (for source address increment) indicates a pattern match.

For 8 bit comparisons, the match conditions LL or LXH or LL AND LXO (LXH is stored in LXO) can be selected (read byte is compared to PAT00 or PAT01 or first PAT01 and then PAT00 in the next move).

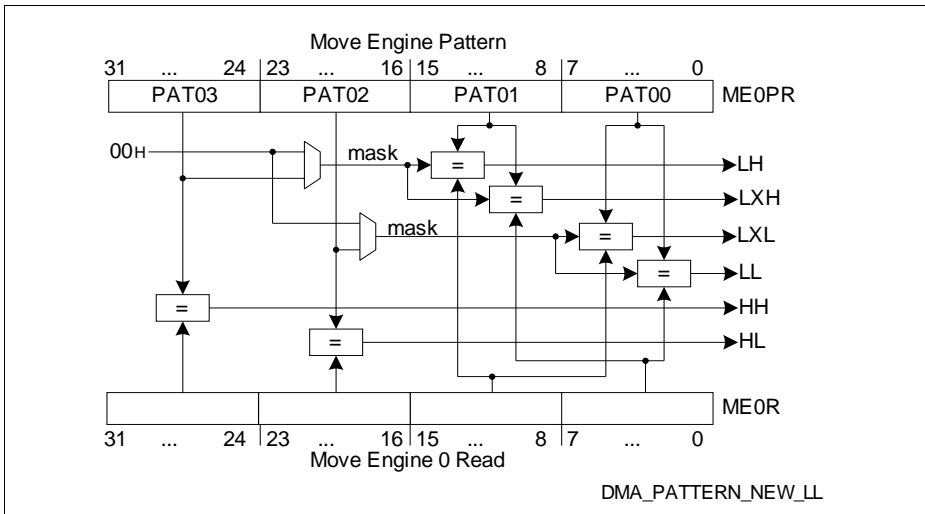


Figure 5-15 Pattern Recognition

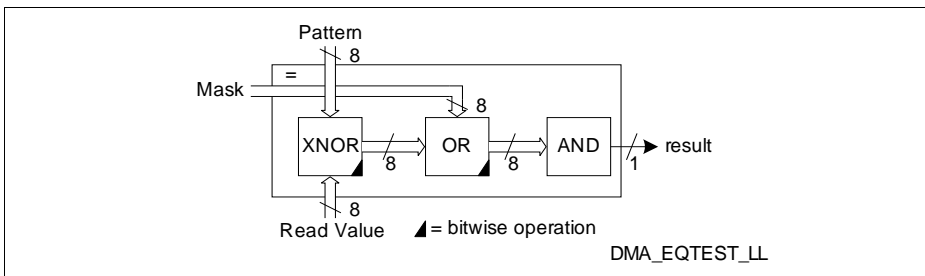


Figure 5-16 Pattern Match Logic

The pattern detection feature can compare the 32 aligned bits, 16 LSB (aligned or not), or only the 8 LSB bits of the data read by the Move Engine, with its pattern.

Direct Memory Access Controller (DMA)

The compare output LXH is stored in CHSR0n.LXHO in order to be used for the following move operation.

The pattern match criteria is met in:

- 32-bit mode when
HH = 1 AND HL = 1 AND LH = 1 AND LL = 1
- 16-bit mode when
LH = 1 AND LL = 1 (pattern aligned)
LXO = 1 AND LXL = 1 (pattern not aligned, source address decrement)
LXO = 1 AND LXH = 1 (pattern not aligned, source address increment)
- 8-bit mode when
LL = 1 (compare to PAT00)
LXH = 1 (compare to PAT01)
LL = 1 AND LXO = 1 (compare to PAT01 first and to PAT00 one move later)

Depending on CHCR0n.PATSEL and on the positive result of the comparison, two actions follow (if CHCR0n.PATSEL=00, no action will be taken when a pattern match is detected, so the wrap interrupt can be used):

- The activation of the interrupt corresponding to the current active channel 0n using the interrupt pointer defined in CHICR0n.WRPP.
- Reset TRSR.HTRE0n and TRSR.CH0n in order to stop the current transaction (hardware and software request enable). The value of CHSR0n.TCOUNT can be read out by the interrupt SW.

The software will have to service the interrupt and to activate again the channel.

5.1.5.9 Error Conditions

The Transaction Lost error flag **ERRSR.TRL0n (n = 0-7)** indicates if a DMA request for a DMA channel 0n has been lost.

In the case of a read error, the write action is not executed, but the destination address is updated.

5.1.5.10 Channel Reset Operation

A DMA transaction of a DMA channel 0n can be stopped (channel is reset) by setting bit **CHRSTR.CH0n (n = 0-7)**. When an Bus access of the DMA channel 0n is just running when setting of CHRSTR.CH0n, this Bus access is finished normally (the running transfer is not aborted). This behavior guarantees data consistency.

When CHRSTR.CH0n is set:

- All rh bits of the corresponding channel are reset when the running transfer is finished correctly. The rw bits of the channel are not changed. Bits TRSR.HTRE0n, TRSR.CH0n, ERRSR.TRL0n, INTSR.ICH0n, INTSR.IPM0n, WRPSR.WRPD0n, WRPSR.WRPS0n, CHSR0n.LXO, CHSR0n.TCOUNT are reset.

Direct Memory Access Controller (DMA)

- If **ADRCR0n (n = 0-7).SHCT** is enabled, the corresponding address will be reloaded with the value in SHADR0n, and the other address will operate a wrap around. This is achieved by setting ADRCR0n.SHCT to 0 (address increment) or setting to 1 (address decrement, except the two least significant bits in order to reach a word boundary) of the address bits that are not “frozen”. If ADRCR0n.SHCT is not enabled both addresses will do a wrap around accordingly.
The address shadow register is cleared.
- All automatic functions are stopped for this channel.

A user program should execute the following steps for resetting and restarting a DMA channel:

1. Set CHRST.CH0n.
2. Poll for CHRST.CH0n = 0.
3. Optionally (re-)configure the address and other channel registers.
4. Restarting the DMA channel 0n again by setting TRSR.HTRE0n or TRSR.CH0n (write a 1 to the corresponding set bit).

The value of bit field CHCR0n.TREL is copied to CHSR0n.TCOUNT when a new transaction is requested.

5.1.5.11 Programmable Address Modification

The DMA is able to modify the source or destination address for each move, taking into account a programmable address modification factor.

The bits in the register ADRCR0n define how the address will be increment/decremented, independently for the source and for the destination buffer. The offset applied on the address is a programmable multiple of the moved data width.

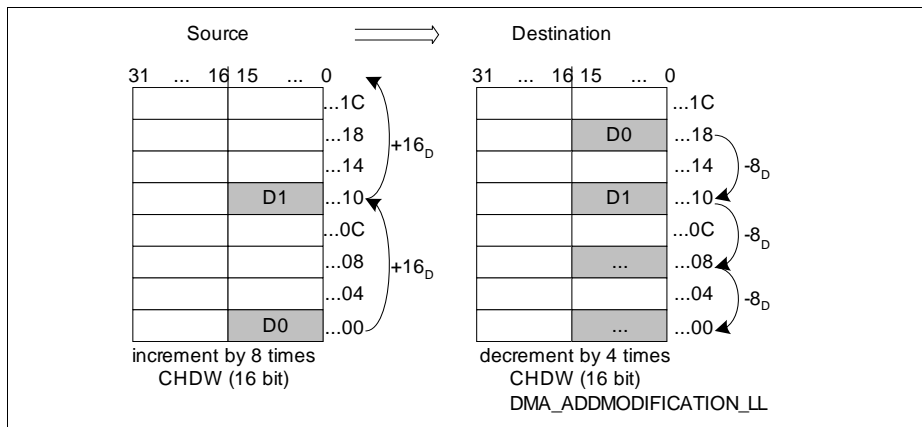


Figure 5-17 Programmable Address Modification

Direct Memory Access Controller (DMA)

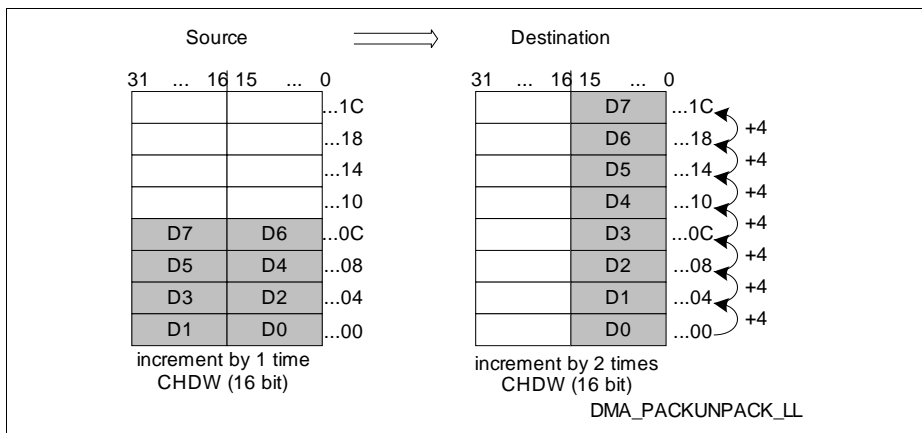


Figure 5-18 Pack / Unpack

Following are the implementation of the address update for source and destination buffer (see [Figure 5-19](#) and [Figure 5-20](#))

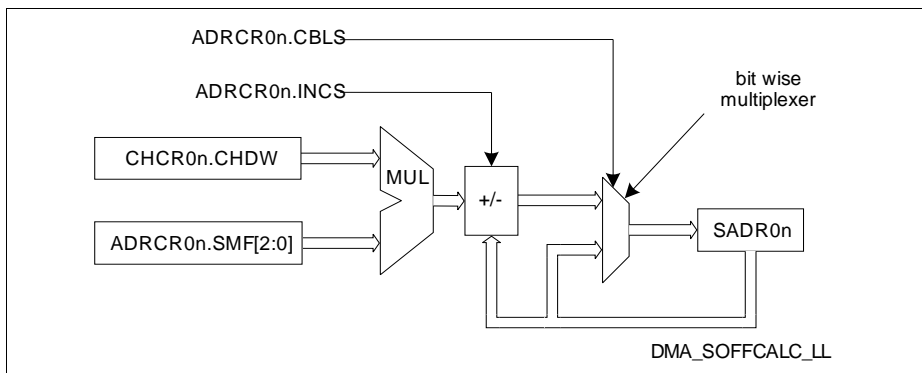


Figure 5-19 Source Address Calculation

Direct Memory Access Controller (DMA)

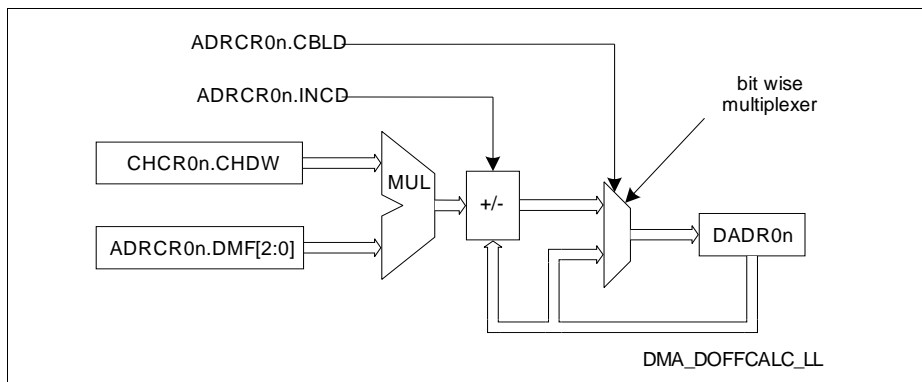


Figure 5-20 Destination Address Calculation

5.1.6 Transaction Control Engine

Each DMA sub-block has a transaction control engine.

The transaction control engine contains a Channel Arbiter and a Move Engine.

The channel arbiter selects the request whose channel has the highest channel priority, and transfers the useful parameters (address for a read, address and data for a write) to the move engine (see [Figure 5-21](#)). The DMA channels of a DMA sub-block have a programmable channel priority (two levels). When two requests come from two different channels with the same channel priority level, the lowest channel number is serviced first. The channel priority is given by the bit **CHCR0n (n = 0-7).CHPRIO**.

The move engine loads and stores data according to the selected channel parameters. The move engine must be able to wait if the targeted bus is not available. Once a DMA transfer (minimum a sequence of one read and one write) has started in a move engine, it must be finished. In other words, the move engine cannot be interrupted in a transfer before selecting a new active channel.

The work registers for the DMA are stored on channel level. Once a DMA transfer is done, the transaction control engine has to update the work registers (source/destination address, transaction counter...) in the channel register set. A move counter is in the move engine, in case there are more than one move for one transfer.

Direct Memory Access Controller (DMA)

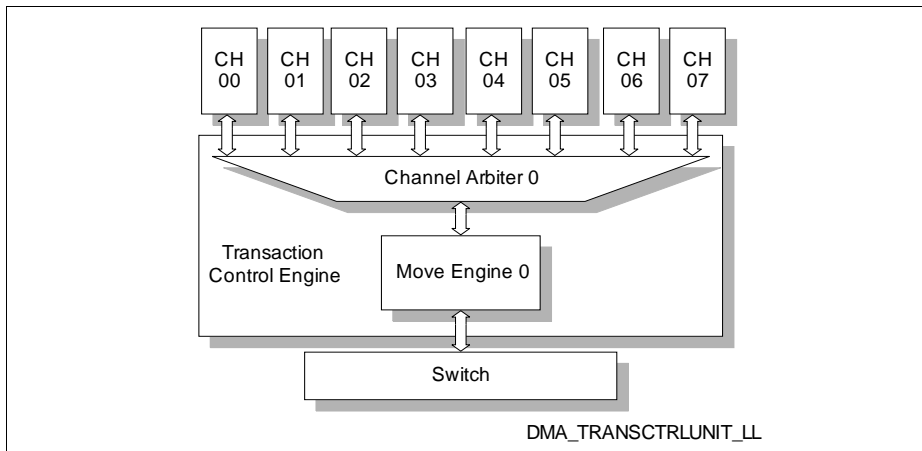


Figure 5-21 Transaction Control Engine

5.1.7 Request Assignment Unit

A DMA sub-block contains one request assignment unit, that multiplexes the request inputs to a single input for each DMA channel.

Each channel can be assigned one out of eight requests. (see [Figure 5-22](#)).

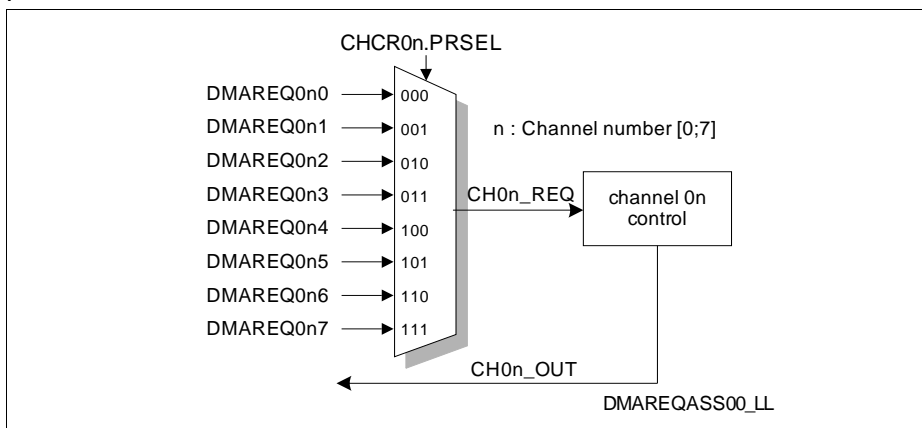


Figure 5-22 Request Assignment Unit for Channel n in Sub-Block 0

Each request input multiplexer of DMA channel 0n is controlled by the peripheral request select bit field CHCR0n.PRSEL. The channel output line ch0n_out is activated when

Direct Memory Access Controller (DMA)

$\text{CHSR0n.TCOUNT} = \text{CHICR0n.IRDV}$ ($\text{CHICR0n.INTCT} = \text{X0}$) or each time CHSR0n.TCOUNT is decremented ($\text{CHICR0n.INTCT} = \text{X1}$).

5.1.8 General Interrupt Structure

The general interrupt structure is shown in [Figure 5-23](#). A interrupt event can trigger the interrupt generation and set the corresponding bit in the status register. The interrupt pulse is generated independently from the interrupt flag in the interrupt status register INTSR. The interrupt flag can be cleared by SW.

If enabled by the related interrupt enable bit, an interrupt pulse can be generated at one of the interrupt output lines int_ox of the module.

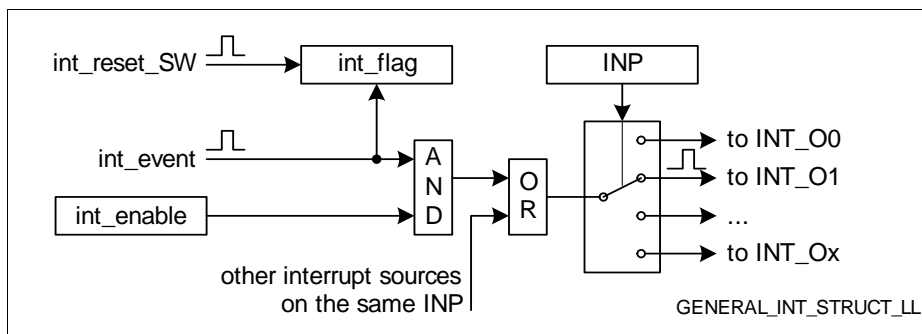


Figure 5-23 General Interrupt Structure

Direct Memory Access Controller (DMA)

5.2 DMA Module Kernel Registers

5.2.1 Overview

Figure 5-24 and Table 5-2 show all registers associated with the DMA Kernel.

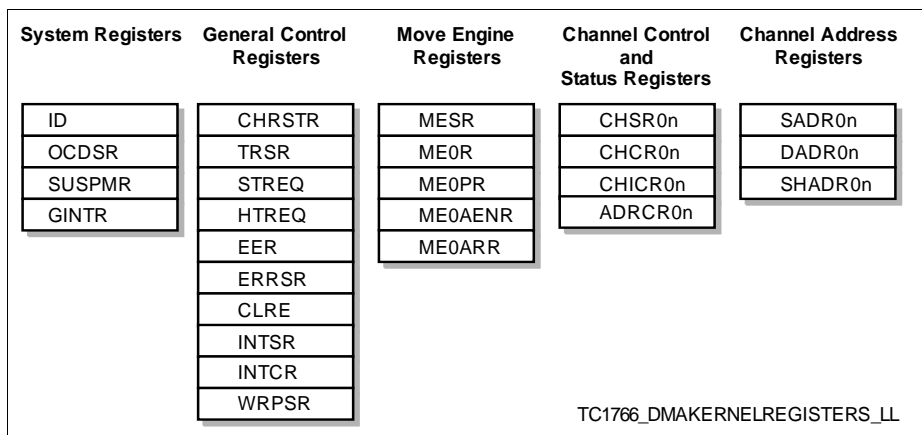


Figure 5-24 DMA Kernel Registers

Table 5-1 DMA Kernel Register Address Space

Module	Base Address	End Address	Note
DMA	0000 0400 _H	0000 05FF _H	256 byte

Direct Memory Access Controller (DMA)
Table 5-2 DMA Kernel Registers

Register Short Name	Register Long Name	Offset Address	Description see
ID	Module Identification Register	0008 _H	Page 5-25
CHRSTR	Channel Reset Request Register	0010 _H	Page 5-27
TRSR	Transaction Request State Register	0014 _H	Page 5-28
STREQ	Software Transaction Request Register	0018 _H	Page 5-29
HTREQ	Hardware Transaction Request Register	001C _H	Page 5-30
EER	Enable Error Register	0020 _H	Page 5-31
ERRSR	Error Status Register	0024 _H	Page 5-33
CLRE	Clear Error Register	0028 _H	Page 5-35
GINTR	Global Interrupt Set Register	002C _H	Page 5-26
MESR	Move Engine Status Register	0030 _H	Page 5-40
ME0R	Move Engine 0 Read Register	0034 _H	Page 5-42
Reserved	-	0038 _H	-
ME0PR	Move Engine 0 Pattern Register	003C _H	Page 5-43
Reserved	-	0040 _H	-
Reserved	-	0044 _H	-
Reserved	-	0048 _H	-
Reserved	-	004C _H	-
Reserved	-	0050 _H	-
INTSR	Interrupt Status Register	0054 _H	Page 5-37
INTCR	Interrupt Clear Register	0058 _H	Page 5-39
WRPSR	Wrap Status Register	005C _H	Page 5-38
Reserved	-	0064 _H	-
Reserved	-	0068 _H	-
CHSR0n ¹⁾	Channel 0n Status Register	$n \times 20_{\text{H}} + 0080_{\text{H}}$	Page 5-49
CHCR0n ¹⁾	Channel 0n Control Register	$n \times 20_{\text{H}} + 0084_{\text{H}}$	Page 5-45
CHICR0n ¹⁾	Channel 0n Interrupt Control Register	$n \times 20_{\text{H}} + 0088_{\text{H}}$	Page 5-50
ADRCR0n ¹⁾	Channel 0n Address Control Register	$n \times 20_{\text{H}} + 008C_{\text{H}}$	Page 5-52

Direct Memory Access Controller (DMA)

Table 5-2 DMA Kernel Registers (cont'd)

Register Short Name	Register Long Name	Offset Address	Description see
SADR0n ¹⁾	Channel 0n Source Address Register	$n \times 20_H + 0090_H$	Page 5-55
DADR0n ¹⁾	Channel 0n Destination Address Register	$n \times 20_H + 0094_H$	Page 5-56
SHADR0n ¹⁾	Channel 0n Shadowed Address Register	$n \times 20_H + 0098_H$	Page 5-57

1) n = 0-7

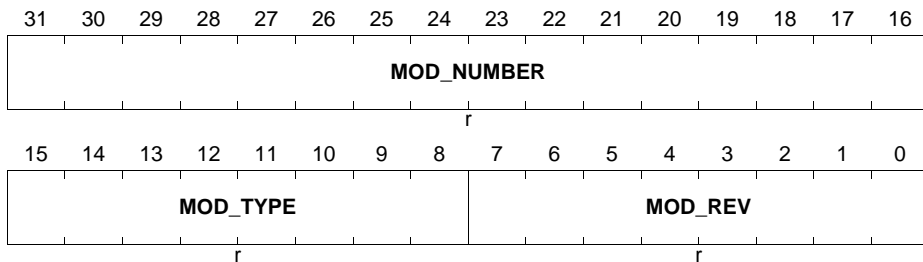
Direct Memory Access Controller (DMA)

5.2.2 Identification Register

5.2.2.1 DMA Module Identification Register

ID

Module Identification Register (0008_H) Reset Value: 001A C012_H



Field	Bits	Type	Description
MOD_REV	[7:0]	r	Module Revision Number MOD_REV defines the module revision number. The current revision is 12 _H .
MOD_TYPE	[15:8]	r	Module Type This bit field is C0 _H . It defines the module as a 32-bit module.
MOD_NUMBER	[31:16]	r	Module Number Value This bit field defines the DMA module identification number 001A _H .

Direct Memory Access Controller (DMA)

5.2.2.2 Global Interrupt Set Register

The Global Interrupt Set Register can activate the service request output signals of the DMA.

GINTR

Global Interrupt Set Register

(002C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI
DMA	DMA	DMA	DMA	DMA	DMA	DMA	DMA	DMA	DMA	DMA	DMA	DMA	DMA	DMA	DMA
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Field	Bits	Type	Description
SIDMA_x (x=0-15)	x	w	Set DMA Interrupt Output Line x 0 No action 1 The DMA Service Request Output line x will be activated.
0	[31:16]	r	Reserved; returns 0 if read; should be written with 0.

Direct Memory Access Controller (DMA)

5.2.3 General Control and Status Registers

5.2.3.1 Channel Reset Request Register

The Channel Reset Request Register resets the desired DMA channels¹.

CHRSTR

Channel Reset Request Register

(0010_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								CH 07	CH 06	CH 05	CH 04	CH 03	CH 02	CH 01	CH 00
r								rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
CH0n (n = 0-7)	n	rwh	Channel 0n Reset These bits force the DMA channel 0n to stop its current DMA transaction. Once set by SW (writing 0 has no effect), this bit will be automatically cleared when the channel has been reset. 0 No action (write) or the requested channel reset has been cleared (read). 1 Stop DMA channel 0n and clear CHSR0n.TCOUNT, TRSR.CH0n, TRSR.HTRE0n, the interrupts flags, the request event counter and the shadowed addresses (write). The requested reset of this channel has not yet been finished (read). More details see Page 5-16 .
0	[31:8]	r	Reserved ; returns 0 if read; should be written with 0.

Direct Memory Access Controller (DMA)

5.2.3.2 Transaction Request State Register

The Transaction Request State Register indicates which DMA channel is processing a request, and which DMA channel has hardware transaction requests enabled.

TRSR

Transaction Request State Register (0014_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								HT RE 07	HT RE 06	HT RE 05	HT RE 04	HT RE 03	HT RE 02	HT RE 01	HT RE 00
r								rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								CH 07	CH 06	CH 05	CH 04	CH 03	CH 02	CH 01	CH 00
r								rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
CH0n (n = 0-7)	n	rh	Transaction Request State of DMA Channel 0n 0 No DMA request is pending for channel 0n. 1 A DMA request is pending for channel 0n.
HTRE0n (n = 0-7)	n + 16	rh	Hardware Transaction Request Enable State of DMA Channel 0n 0 Hardware transaction request for DMA Channel 0n is disabled. An input DMA request will not trigger the channel 0n. 1 Hardware transaction request for DMA Channel 0n is enabled. The transfers of a DMA transaction are controlled by the corresponding channel request line of the DMA requesting source. HTRE0n is set to 0 when CHSR0n.TCOUNT is decremented and CHSR0n.TCOUNT = 0. HTRE0n can be enabled and disabled with HTREQ.ECH0n or HTREQ.DCH0n.
0	[15:8], [31:24]	r	Reserved ; returns 0 if read; should be written with 0.

Direct Memory Access Controller (DMA)

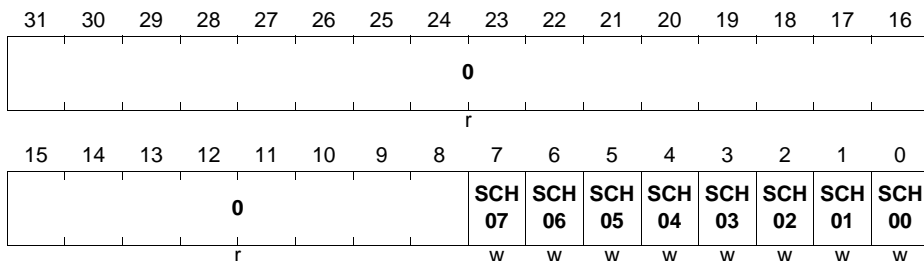
5.2.3.3 Software Transaction Request Register

The Software Transaction Request Register can trigger a DMA request by SW.

STREQ

Software Transaction Request Register(0018_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
SCH0n (n = 0-7)	n	w	Set Transaction Request for DMA Channel 0n 0 No action. 1 Bit TRSR.CH0n will be set.
0	[31:8]	r	Reserved ; returns 0 if read; should be written with 0.

Direct Memory Access Controller (DMA)

5.2.3.4 Hardware Transaction Request Register

The Hardware Transaction Request Register enables or disables DMA hardware requests.

HTREQ

Hardware Transaction Request Register(001C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								DCH 07	DCH 06	DCH 05	DCH 04	DCH 03	DCH 02	DCH 01	DCH 00
r								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								ECH 07	ECH 06	ECH 05	ECH 04	ECH 03	ECH 02	ECH 01	ECH 00
r								w	w	w	w	w	w	w	w

Field	Bits	Type	Description
ECH0n (n = 0-7)	n	w	Enable Hardware Transfer Request for DMA Channel 0n See table below
DCH0n (n = 0-7)	n + 16	w	Disable Hardware Transfer Request for DMA Channel 0n See table below
0	[15:8], [31:24]	r	Reserved ; returns 0 if read; should be written with 0.

Table 5-3 Conditions to Set/Reset the Bit TRSR.HTREQ0n

HTREQ.ECH0n	HTREQ.DCH0n	Transaction finishes ¹⁾ for channel 0n	Modification of TRSR.HTREQ0n
0	0	0	Unchanged
1	0	0	Set
X	1	X	Cleared
X	X	1	Cleared

1) In Single Mode only. In Continuous Mode, the end of a transaction has no impact.

Direct Memory Access Controller (DMA)

5.2.3.5 Enable Error Register

The Enable Error Register describes how the DMA controller reacts to errors. It enables the interrupts for the loss of a transaction request or move engine errors.

EER

Enable Error Register

(0020_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRLINP				0				ME0INP				0		E ME0 DER	E ME0 SER
rw				r				rw				r		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								E TRL 07	E TRL 06	E TRL 05	E TRL 04	E TRL 03	E TRL 02	E TRL 01	E TRL 00
r								rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
ETRL0n (n = 0-7)	n	rw	Enable Transaction Request Lost for DMA Channel 0n This bit enables the generation of an interrupt when the set condition for ERRSR.TRL0n is detected. 0 The interrupt generation for a request lost event for channel 0n is disabled. 1 The interrupt generation for a request lost event for channel 0n is enabled.
EME0SER	16	rw	Enable Move Engine 0 Source Error This bit enables the generation of an interrupt when the set condition for ERRSR.ME0SER is detected. 0 The interrupt generation for this event is disabled. 1 The interrupt generation for this event is enabled.
EME0DER	17	rw	Enable Move Engine 0 Destination Error This bit enables the generation of an interrupt when the set condition for ERRSR.ME0DER is detected. 0 The interrupt generation for this event is disabled. 1 The interrupt generation for this event is enabled.

Direct Memory Access Controller (DMA)

Field	Bits	Type	Description
ME0INP	[23:20]	rw	Move Engine 0 Error Interrupt Node Pointer References the service request node to be set when an error occurs in move engine 0.
TRLINP	[31:28]	rw	Transaction Lost Interrupt Node Pointer References the service request node to be set when a transaction request is lost
0	[15:8], [19:18], [27:24]	r	Reserved ; returns 0 if read; should be written with 0.

Direct Memory Access Controller (DMA)

5.2.3.6 Error Status Register

The Error Status Register indicates if the DMA controller couldn't answer to a request because the previous request was not terminated (see [Section 5.1.5.9](#)).

ERRSR

Error Status Register

(0024_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
XS	0			MLI	LEC ME			0		BER 1	BER 0	0		ME0 DER	ME0 SER
rh	r			rh	rh			r		rh	rh	r		rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								TRL 07	TRL 06	TRL 05	TRL 04	TRL 03	TRL 02	TRL 01	TRL 00
r								rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
TRL0n (n = 0-7)	n	rh	Transaction/Transfer Request Lost of DMA Channel 0n 0 No request lost event has been detected for channel 0n. 1 A new DMA request was detected while TRSR.CH0n=1 (request lost event).
ME0SER	16	rh	Move Engine 0 Source Error This bit is set whenever a move engine 0 error occurred during a source (read) move of a DMA transfer or a request could not been serviced due to the access protection. 0 No move engine 0 source error has occurred. 1 A move engine 0 source error has occurred.
ME0DER	17	rh	Move Engine 0 Destination Error This bit is set whenever a move engine 0 error occurred during a destination (write) move of a DMA transfer or a request could not been serviced due to the access protection. 0 No move engine 0 destination error has occurred. 1 A move engine 0 destination error has occurred.

Direct Memory Access Controller (DMA)

Field	Bits	Type	Description
BER0	20	rh	Bus Error 0 This bit is set whenever a move that has been started by the DMA/MLI/SSC/XMU master interface accessing the ports, SCU or SSC lead to an error. 0 No error on the bus has occurred. 1 An error on the bus has occurred.
BER1	21	rh	Bus Error 1 This bit is set whenever a move that has been started by the DMA/MLI/SSC/XMU master interface on the accessing the ERAY lead to an error. 0 No error on the bus has occurred. 1 An error on the bus has occurred.
LECME	[26:24]	rh	Last Error Channel Move Engine This bit field indicates the channel number of the last channel of move engine leading to an Bus error occurred.
MLI	27	rh	MLI Error Source This bit is set whenever an Bus error occurred due to an action of MLI. 0 No Bus error occurred due to MLI. 1 An Bus error occurred due to MLI.
XS	31	rh	SSC and XMU Error Source This bit is set whenever an Bus error occurred due to an bus master action of SSC or XMU. 0 No Bus error occurred due to SSC or XMU bus master access. 1 An Bus error occurred due to SSC or XMU bus master access.
0	[15:8], [19:18], [23:22], [30:28]	r	Reserved ; returns 0 if read; should be written with 0.

Direct Memory Access Controller (DMA)

5.2.3.7 Clear Error Register

The Clear Error clears the Transaction Request Lost bit or the move engine error indications.

CLRE

Clear Error Register

(0028_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLR XS	0			CLR MLI	0			C BER 1			C BER 0	0		C ME0 DER	C ME0 SER
w	r			w	r			w			w	r		w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								C TRL 07	C TRL 06	C TRL 05	C TRL 04	C TRL 03	C TRL 02	C TRL 01	C TRL 00
r								w	w	w	w	w	w	w	w

Field	Bits	Type	Description
CTRL0n (n = 0-7)	n	w	Clear Transaction Request Lost for the DMA Channel 0n 0 No action 1 Clear DMA channel 0n transaction request lost flag ERRSR.TRL0n
CME0SER	16	w	Clear Move Engine 0 Source Error 0 No action 1 Clear source error flag ERRSR.ME0SER.
CME0DER	17	w	Clear Move Engine 0 Destination Error 0 No action 1 Clear destination error flag ERRSR.ME0DER.
CBER0	20	w	Clear Bus Error 0 0 No action 1 Clear destination error flag ERRSR.BERR0.
CBER1	21	w	Clear Bus Error 1 0 No action 1 Clear destination error flag ERRSR.BERR1.

Direct Memory Access Controller (DMA)

Field	Bits	Type	Description
CLRMLI	27	w	Clear MLI Error 0 No action 1 Clear error flag ERRSR.MLI.
CLRXS	31	w	Clear SSC and XMU Error 0 No action 1 Clear error flag ERRSR.XS.
0	[15:8], [19:18], [26:22], [30:28]	r	Reserved ; returns 0 if read; should be written with 0.

Direct Memory Access Controller (DMA)

5.2.3.8 Interrupt Status Register

The Interrupt Status Register indicates if CHSR0n.TCOUNT=CHCR0n.IRDV has been detected or if CHSR0n.TCOUNT has been decremented (depending on CHICR0n.INTCT.0). A pattern match is also indicated by this register. These conditions can also lead to interrupts if enabled.

INTSR

Interrupt Status Register

(0054_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								IPM 07	IPM 06	IPM 05	IPM 04	IPM 03	IPM 02	IPM 01	IPM 00
r								rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								ICH 07	ICH 06	ICH 05	ICH 04	ICH 03	ICH 02	ICH 01	ICH 00
r								rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
ICH0n (n = 0-7)	n	rh	Interrupt from Channel 0n This bit indicates that channel 0n has raised an interrupt for TCOUNT = IRDV or if TCOUNT has been decremented (depending on CHICR.INTCT.0). It can be reset by SW by writing a 1 to INTCR.CICH0n or by a channel reset. 0 A channel interrupt event has not been detected. 1 A channel interrupt event has been detected.
IPM0n (n = 0-7)	n+ 16	rh	Pattern Detection from Channel 0n This bit indicates that a pattern has been detected for channel 0n while the pattern detection has been enabled. It can be reset by SW by writing a 1 to INTCR.CICH0n or by a channel reset. 0 A pattern has not been detected. 1 A pattern has been detected and a trigger for a channel interrupt has been generated.
0	[15:8], [31:24]	r	Reserved; returns 0 if read; should be written with 0.

Direct Memory Access Controller (DMA)

5.2.3.9 Wrap Status Register

The Wrap Status Register gives information about the channels that did a wrap around on their source or destination buffer. This condition can also lead to an interrupt if it is enabled.

WRPSR

Wrap Status Register

(005C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								WRP D07	WRP D06	WRP D05	WRP D04	WRP D03	WRP D02	WRP D01	WRP D00
r								rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								WRP S07	WRP S06	WRP S05	WRP S04	WRP S03	WRP S02	WRP S01	WRP S00
r								rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
WRPS0n (n = 0-7)	n	rh	Wrap Source Buffer for Channel 0n These bits indicate which channels have done a wrap around of their source buffer. 0 No wrap around occurred for channel 0n. 1 A wrap around occurred for channel 0n. This bit is reset by SW by writing a 1 to INTCR.CWRP0n.
WRPD0n (n = 0-7)	n +16	rh	Wrap Destination Buffer for Channel 0n These bits indicate which channels have done a wrap around of their destination buffer. 0 No wrap around occurred for channel 0n. 1 Wrap around occurred for channel 0n. This bit is reset by SW by writing a 1 to INTCR.CWRP0n.
0	[15:8], [31:24]	r	Reserved ; returns 0 if read; should be written with 0.

Direct Memory Access Controller (DMA)

5.2.3.10 Interrupt Clear Register

The Interrupt Clear Register reset the interrupt flags of the DMA Channels.

INTCR

Interrupt Clear Register

(0058_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								CWR P07	CWR P06	CWR P05	CWR P04	CWR P03	CWR P02	CWR P01	CWR P00
r								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								CICH 07	CICH 06	CICH 05	CICH 04	CICH 03	CICH 02	CICH 01	CICH 00
r								w	w	w	w	w	w	w	w

Field	Bits	Type	Description
CICH0n (n = 0-7)	n	w	Clear Interrupt for Channel 0n These bits clear the interrupt flags for channel mn (INTSR.ICH0n and INTSR.IPM0n). 0 Bits INTSR.ICH0n and INTSR.IPM0n are not changed. 1 Bit INTSR.ICH0n and INTSR.IPM0n are reset.
CWRP0n (n = 0-7)	n + 16	w	Clear Wrap Indication for Channel 0n These bits clear both (source/destination) wrap indication for channel 0n in WRPSR register. 0 Bits WRPSR.S0n and WRPSR.D0n are not changed by SW. 1 Bits WRPSR.S0n and WRPSR.D0n are reset.
0	[15:8], [31:24]	r	Reserved ; returns 0 if read; should be written with 0.

Direct Memory Access Controller (DMA)

5.2.4 Move Engine Registers

5.2.4.1 Move Engine Status Register

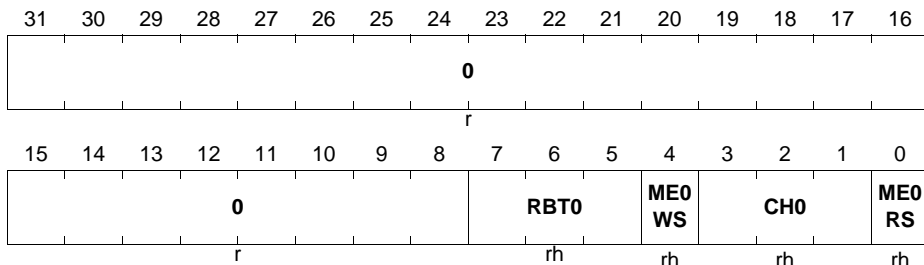
The move engine status register gives information about the transaction treated by the move engine.

MESR

Move Engine Status Register

(0030_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
ME0RS	0	rh	Move Engine 0 Read Status 0 Move engine 0 is not doing a read. 1 Move engine 0 is doing a read.
CH0	[3:1]	rh	Reading Channel in Move Engine 0 These bit fields indicate which channel number is currently being processed by the move engine 0.
ME0WS	4	rh	Move Engine 0 Write Status 0 Move engine 0 is not doing a write. 1 Move engine 0 is doing a write.

Direct Memory Access Controller (DMA)

Field	Bits	Type	Description
RBT0	[7:5]	rh	Read Buffer Trace for Bus Access This bit field contains the trace information from the buffer in the DMA master interface on Bus. It indicates the source of a bus access to the corresponding bus. This information can only be used together with a cycle-accurate trace of the Bus activities (can done by an emulation device). Analyzing this bit field under program control will not always give the full information. 000 _B Reset value 001 _B DMA move engine 0 011 _B MLI 100 _B SSC Else Reserved
0	[31:8]	r	Reserved ; returns 0 if read; should be written with 0.

Direct Memory Access Controller (DMA)

5.2.4.2 Move Engine 0 Read Register

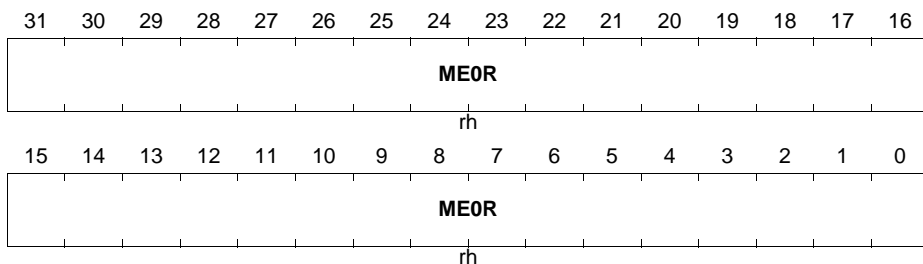
The move engine 0 read register indicates the value that has just been read by the move engine 0. The value in this register is compared to the bits in register ME0PR according to the bit fields CHCR0n.PATSEL.

ME0R

Move Engine 0 Read Register

(0034_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
ME0R	[31:0]	rh	Move Engine 0 Read Value Contains the 32-bit value stored in the move engine after a read.

Note: The result of an 8 bit wide read is copied to the locations ME0R[7:0], ME0R[15:8], ME0R[23:16], and ME0R[31:24]. The result of a 16 bit wide read is copied to the locations ME0R[15:0] and ME0R[31:16]. The result of a 32 bit wide read is copied to the location ME0R[31:0].

Direct Memory Access Controller (DMA)

5.2.4.3 Move Engine 0 Pattern Register

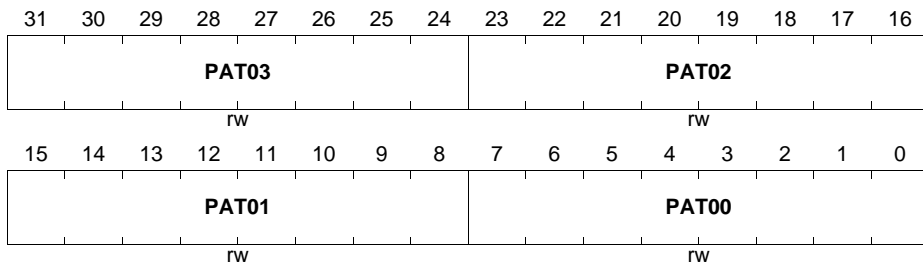
The move engine 0 pattern register defines the patterns to be searched for by the move engine 0.

ME0PR

Move Engine 0 Pattern Register

(003C_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
PAT00	[7:0]	rw	Pattern for Move Engine 0 Defines the four 8-bit pattern to be compared with the data read by the move engine for a DMA channel. Depending on how the channel is configured in CHCR00.PATSEL, the pattern can be assembled to 16 bits or 32 bits, for word pattern recognition
PAT01	[15:8]	rw	Pattern for Move Engine 0 Defines the four 8-bit pattern to be compared with the data read by the move engine for a DMA channel. Depending on how the channel is configured in CHCR01.PATSEL, the pattern can be assembled to 16 bits or 32 bits, for word pattern recognition
PAT02	[23:16]	rw	Pattern for Move Engine 0 Defines the four 8-bit pattern to be compared with the data read by the move engine for a DMA channel. Depending on how the channel is configured in CHCR02.PATSEL, the pattern can be assembled to 16 bits or 32 bits, for word pattern recognition

Direct Memory Access Controller (DMA)

Field	Bits	Type	Description
PAT03	[31:24]	rw	Pattern for Move Engine 0 Defines the four 8-bit pattern to be compared with the data read by the move engine for a DMA channel. Depending on how the channel is configured in CHCR03.PATSEL, the pattern can be assembled to 16 bits or 32 bits, for word pattern recognition

Direct Memory Access Controller (DMA)

Field	Bits	Type	Description
BLKM	[18:16]	rw	Block Mode Defines the number of moves to be done for each transaction request. 000 _B 1 move 001 _B 2 moves 010 _B 4 moves 011 _B 8 moves 100 _B 16 moves else Reserved: Should not be used.
RROAT	19	rw	Reset Request Only After Transfer 0 Reset of TRSR.CH0n after each transfer. A trigger event is required for each transfer. 1 Reset of TRSR.CH0n each time TCOUNT = 0 after a transfer. In this mode, one trigger event leads to a complete transfer.
CHMODE	20	rw	Channel Operation Mode This bit field defines the operating mode of DMA channel 0n. 0 The Single Mode operation is selected for DMA channel 0n. After a transaction, bit TRSR.HTRE0n is automatically cleared. In this mode, the DMA request inputs are only taken into account until the transaction is finished. In order to continue transfers triggered by DMA request inputs, TRSR.HTRE0n has to be set again. 1 The Continuous Mode operation is selected for DMA channel 0n. After a transaction, bit TRSR.HTRE0n is not changed. In this mode, the DMA request inputs can permanently trigger DMA transfers or transactions (depending on the bit field CHCR0n.RROAT).
CHDW	[22:21]	rw	Channel Data Width CHDW specifies the data width for source and destination transactions of DMA channel 0n. 00 _B 8-bit (byte) transaction selected 01 _B 16-bit (half-word) transaction selected 10 _B 32-bit (word) transaction selected 11 _B Reserved: Should not be used.

Direct Memory Access Controller (DMA)

Field	Bits	Type	Description
PATSEL	[25:24]	rw	<p>Pattern Select</p> <p>This bit field selects the mode of the pattern comparison and enables the interrupt generation for pattern matches. The pattern match should not be enabled while a wrap interrupt is enabled for the same channel. A positive pattern match is indicated by the status bit for the wrap around of the source address pointer.</p> <ul style="list-style-type: none"> • If CHDW = 00: (8 bit move) <ul style="list-style-type: none"> 00_B The pattern detection and corresponding interrupt generation are disabled. 01_B Only compare with pattern PAT00 match criteria: LL. 10_B Only compare with pattern PAT01 match criteria: LXH. 11_B Compare with pattern PAT01 first and PAT00 one move later match criteria: LL and LXO. • If CHDW = 01: (16 bit move) <ul style="list-style-type: none"> 00_B The pattern detection and corresponding interrupt generation are disabled. 01_B Compare with pattern [PAT01:PAT00], aligned match criteria: LL and LH. 10_B Compare with pattern [PAT01:PAT00], not aligned, match criteria: LXL and LXO (decrement) or LXH and LXO (increment). 11_B Compare with pattern [PAT01:PAT00], not aligned or aligned match criteria: LL and LH or LXL and LXO (decrement) or LXH and LXO (increment) • If CHDW = 10 or 11: (32 bit move) <ul style="list-style-type: none"> 00_B The pattern detection and corresponding interrupt generation are disabled. 01_B Only compare with pattern [PAT01:PAT00], aligned, match criteria: LH and LL. 10_B Only compare with pattern [PAT03:PAT02], aligned, match criteria: HH and HL. 11_B Compare with the pattern [PAT03:PAT0PAT00], match criteria: HH and HL and LH and LL

Direct Memory Access Controller (DMA)

Field	Bits	Type	Description
CHPRIO	28	rw	Channel Priority This bit defines the priority of the channels in the channel arbitration for the move engine. 0 Low priority 1 High priority
0	30	rw	Reserved ; reading these bits will return the value last written; read as 0 after reset.
0	[12:9], 23, [27:26], 31,29	r	Reserved ; returns 0 if read; should be written with 0.

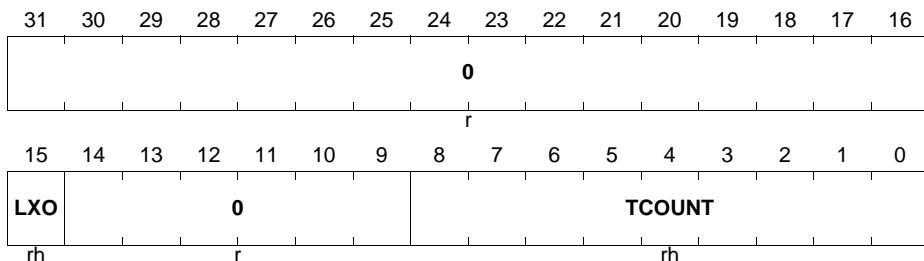
Direct Memory Access Controller (DMA)

5.2.5.2 Channel 0n Status Register (n = 0-7)

The channel status register assigned to each DMA channel contains its status flags.

CHSR0n (n = 0-7)

Channel 0n Status Register (n = 0-7)(n * 20_H + 80_H) **Reset Value: 0000 0000_H**



Field	Bits	Type	Description
TCOUNT	[8:0]	rh	Transfer Count Status This bit field contains the actual value of the DMA channel 0n transfer count. After completion of a DMA transfer, the bit field CHSR0n.TCOUNT is decremented by 1. It is set to the value of CHCR0n.TREL when TCOUNT=0 and TRSR.CH0n becomes set.
LXO	15	rh	Old Value of LXH/LXL This bit contains the comparison result LXH/LXL of the previous read action. <ul style="list-style-type: none"> 8 bit moves: LXH is stored in LXO 16 bit moves: LXH is stored in LXO if the source address is selected to be decremented or LXL is stored in LXO if the source address is selected to be increment 32 bit moves: like 16 bit moves 0 The corresponding compare action didn't deliver a pattern match for the last move. 1 The corresponding compare action delivered a pattern match for the last move.
0	[14:9], [31:16]	r	Reserved; returns 0 if read; should be written with 0.

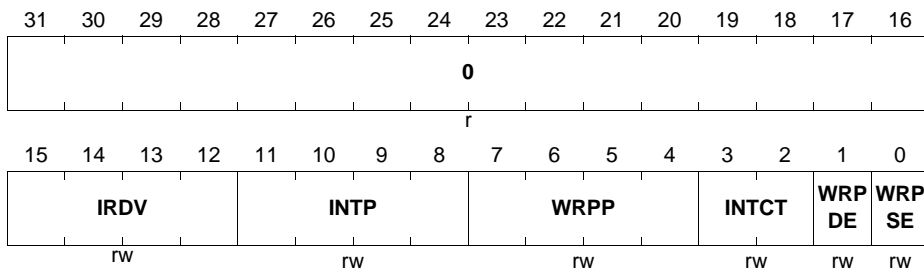
Direct Memory Access Controller (DMA)

5.2.5.3 Channel Interrupt Control Register (n = 0-7)

The channel interrupt control register control the interrupts generation.

CHICR0n (n = 0-7)

Channel 0n Interrupt Control (n = 0-7)(n * 20_H + 88_H) Reset Value: 0000 0000_H



Field	Bits	Type	Description
WRPSE	0	rw	Wrap Source Enable 0 Wrap source interrupt disabled 1 Wrap source interrupt enabled
WRPDE	1	rw	Wrap Destination Enable 0 Wrap destination interrupt disabled 1 Wrap destination interrupt enabled
INTCT	[3:2]	rw	Interrupt Control 00 _B No interrupt will be generated on changing the TCOUNT value. The bit INTSR.ICH0n is set when TCOUNT equals IRDV. 01 _B No interrupt will be generated on changing the TCOUNT value. The bit INTSR.ICH0n is set when TCOUNT is decremented. 10 _B An interrupt is generated and bit INTSR.ICH0n is set each time TCOUNT equals IRDV. 11 _B Interrupt is generated and bit INTSR.ICH0n is set each time TCOUNT is decremented. (see Figure 5-11)
WRPP	[7:4]	rw	Wrap Pointer References the service request node to be set if a wrap around source or destination address occurs (while the corresponding wrap interrupt is enabled).

Direct Memory Access Controller (DMA)

Field	Bits	Type	Description
INTP	[11:8]	rw	Interrupt Pointer References the service request node to be set when CHSR0n.TCOUNT = CHICR0n.IRDV or when TCOUNT is decremented (according to INTCT). A pattern detection interrupt also uses this node pointer.
IRDV	[15:12]	rw	Interrupt Raise Detect Value These bits specifies the value of CHSR0n.TCOUNT for which the interrupt threshold limit should be raised.
0	[31:16]	r	Reserved ; returns 0 if read; should be written with 0.

Note: The interrupt node of the channel interrupt is shared with the pattern match interrupt. In order to support interrupt generation in case of a pattern match, the channel interrupt on TCOUNT should be disabled.

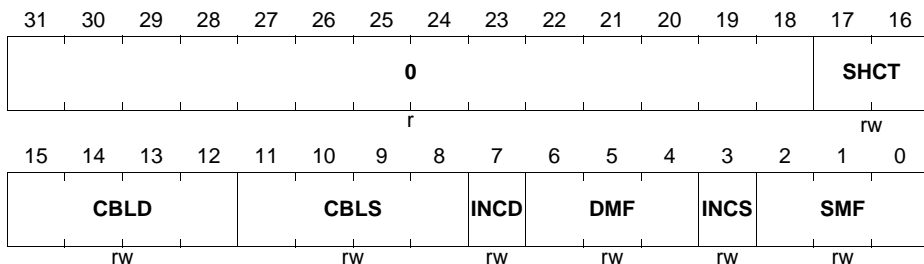
Direct Memory Access Controller (DMA)

5.2.5.4 Address Control Register (n = 0-7)

The address control register control the way the address is modified after each move.

ADRCR0n (n = 0-7)

Channel 0n Address Control Register(n * 20H + 8C_H) **Reset Value: 0000 0000_H**



Field	Bits	Type	Description
SMF	[2:0]	rw	Source Modification Factor This bit field defines the factor by which the source address will be modified (increment or decrement, depending on INCS) after each move operation. The source address is not modified if CBLS = 0000. 000 _B The update factor is one time the data width. 001 _B The update factor is 2 times the data width. 010 _B The update factor is 4 times the data width. 011 _B The update factor is 8 times the data width. 100 _B The update factor is 16 times the data width. 101 _B The update factor is 32 times the data width. 110 _B The update factor is 64 times the data width. 111 _B The update factor is 128 times the data width.
INCS	3	rw	Increment of Source Address This bit defines if the source address will be increment or decrement after each move operation. The source address is not modified if CBLS = 0000. 0 The source address will be decremented. 1 The source address will be increment.

Direct Memory Access Controller (DMA)

Field	Bits	Type	Description
DMF	[6:4]	rw	Destination Modification Factor This bit field defines the factor by which the destination address will be modified (increment or decrement, depending on INCD) after each move operation. The destination address is not modified if CBLD = 0000. 000 _B The update factor is one time the data width. 001 _B The update factor is 2 times the data width. 010 _B The update factor is 4 times the data width. 011 _B The update factor is 8 times the data width. 100 _B The update factor is 16 times the data width. 101 _B The update factor is 32 times the data width. 110 _B The update factor is 64 times the data width. 111 _B The update factor is 128 times the data width.
INCD	7	rw	Increment of Destination Address This bit defines if the destination address will be increment or decrement after each move operation. The destination address is not modified if CBLD = 0000. 0 The destination address will be decremented. 1 The destination address will be increment.
CBLD	[11:8]	rw	Circular Buffer Length Source This 4 bit binary value indicates the first bit position in the 32-bit source address register that is left unchanged after a move operation (see also Section 5.1.5.6). 0000 _B The address bits SADR[31:0] are left unchanged. The address is not modified. 0001 _B The address bits SADR[31:1] are left unchanged. This corresponds to a circular buffer of 2 byte. 0010 _B The address bits SADR[31:2] are left unchanged. This corresponds to a circular buffer of 4 byte. 0011 _B The address bits SADR [31:3] are left unchanged. This corresponds to a circular buffer of 8 byte. ... 1110 _B The address bits SADR[31:14] are left unchanged. This corresponds to a circular buffer of 16 kbyte. 1111 _B The address bits SADR[31:15] are left unchanged. This corresponds to a circular buffer of 32 kbyte.

Direct Memory Access Controller (DMA)

Field	Bits	Type	Description
CBLD	[15:12]	rw	Circular Buffer Length Destination This 4 bit binary value indicates the first bit position in the 32-bit destination address register that is left unchanged after a move operation (see also Section 5.1.5.6). 0000 _B The address bits DADR[31:0] are left unchanged. The address is not modified. 0001 _B The address bits DADR[31:1] are left unchanged. This corresponds to a circular buffer of 2 byte. 0010 _B The address bits DADR[31:2] are left unchanged. This corresponds to a circular buffer of 4 byte. 0011 _B The address bits DADR[31:3] are left unchanged. This corresponds to a circular buffer of 8 byte. ... 1110 _B The address bits DADR[31:14] are left unchanged. This corresponds to a circular buffer of 16 Kbyte. 1111 _B The address bits DADR[31:15] are left unchanged. This corresponds to a circular buffer of 32 Kbytes.
SHCT	[17:16]	rw	Shadow Control 00 _B Shadow register not used 01 _B Shadow register used for Source 10 _B Shadow register used for Destination 11 _B Reserved: Should not be used
0	[31:18]	r	Reserved ; returns 0 if read; should be written with 0.

Note: The bit fields SMF and DMF together with CHCR0n.CHDW are defining the value by which the corresponding address can be updated after each move operation. With a selected data width of 8 bits (1 byte), the next move can target addresses that are 1, 2, 4, 8, 16, 32, 64 or 128 byte before or after the last move address. With a selected data width of 16 bits (2 byte), the next move can target addresses that are 2, 4, 8, 16, 32, 64, 128 or 256 byte before or after the last move address. With a selected data width of 32 bits (4 byte), the next move can target addresses that are 4, 8, 16, 32, 64, 128, 256 or 512 byte before or after the last move address. The calculated addresses are then mapped to the circular buffer, where the upper calculated address bits are not taken into account if a smaller buffer size is selected.

Direct Memory Access Controller (DMA)

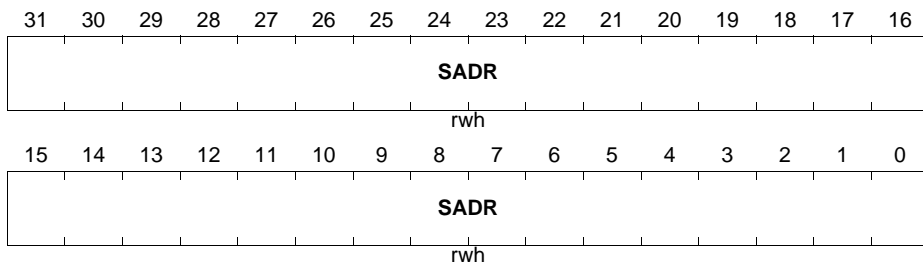
5.2.5.5 Source Address Register (n = 0-7)

The source address register contains the 32-bit start address of the source buffer.

SADR0n (n = 0-7)

Channel 0n Source Address (n = 0-7)(n*20_H+90_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
SADR	[31:0]	rwh	Source Start Address This bit field specifies the 32-bit address of the source buffer of Channel 0n.

To write SADR0n.SADR the corresponding channel must be inactive (CHSR0n.TCOUNT = 0 and TRSR.CH0n = 0).

If the channel is active when writing and ADRCR0n.SHCT = 01 then the address will be written in the address shadow register.

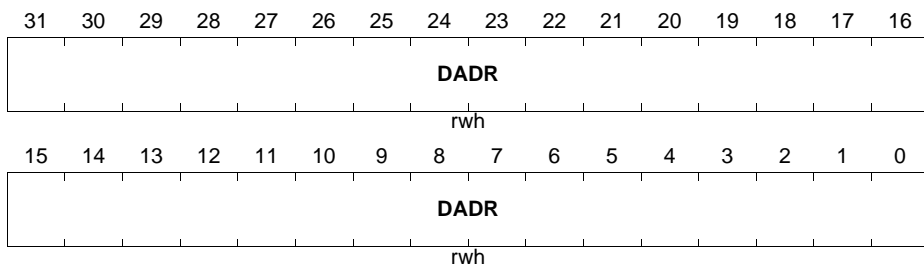
Direct Memory Access Controller (DMA)

5.2.5.6 Destination Address Register (n = 0-7)

The destination address register contains the 32-bit start address of the destination buffer.

DADR0n (n = 0-7)

Channel 0n Destination Address (n = 0-7)(n * 20_H + 94_H) Reset Value: 0000 0000_H



Field	Bits	Type	Description
DADR	[31:0]	rwh	Destination Address This bit field specifies the 32-bit address of the destination buffer of DMA Channel 0n.

To write DADR0n.DADR the corresponding channel must be inactive (CHSR0n.TCOUNT = 0 and TRSR.CH0n = 0).

If the channel is active when writing and ADRCR0n.SHCT = 10 then the address will be written in the address shadow register.

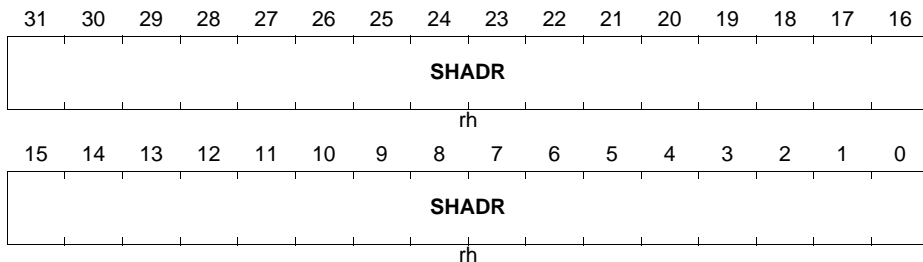
Direct Memory Access Controller (DMA)

5.2.5.7 Shadow Address Register (n = 0-7)

The shadow address register contains the source or destination shadowed address.

SHADR0n (n = 0-7)

Channel 0n Shadow Address (n = 0-7)(n * 20_H + 0098_H) **Reset Value: 0000 0000_H**



Field	Bits	Type	Description
SHADR	[31:0]	rh	Shadowed Address This bit field specifies the 32-bit address in the shadowed address buffer of DMA Channel 0n.

To write SHADR0n, ADRCR0n.SHCT must be activated (ADRCR0n.SHCT = 01 or ADRCR0n.SHCT = 10), a transaction must be running and the write address must be SADR0n or DADR0n (depending on ADRCR0n.SHCT). While the shadow mechanism is disabled, SHADR is set to 0000 0000_H.

The value stored in the shadow register is automatically set to 0000 0000_H when the shadow transfer takes place. The user can read the shadow register in order to detect if the shadow transfer has already taken place.

If the value in the shadow register is 0000 0000_H then no shadow transfer can take place and the corresponding address register is modified according to the circular buffer rules.

If both address registers (for source and for the destination address) have to be configured, the last (current) transaction for this channel must be finished completely. Only one address register can be re-configured while a transaction is running, because the shadow register can only be assigned either to the source or to the destination address register. While a transaction is currently not running, the value of ADRCR0n.SHCT is not taken into account (the write access takes place directly to the register).

Direct Memory Access Controller (DMA)

5.3 DMA Module Implementation

5.3.1 DMA Request Assignment Matrix

The DMA request input lines of the DMA according to the following table:

Table 5-4 DMA Request Assignment

DMA Channel	DMA Request Input	DMA Request Source	Selected by
00	ssc_tir	SSC Transmit Interrupt	CHCR00.PRSEL=000 _B
	ssc_rir	SSC Receive Interrupt	CHCR00.PRSEL=001 _B
	mli2_o	MLI	CHCR00.PRSEL=010 _B
	mli3_o	MLI	CHCR00.PRSEL=011 _B
	rt	Ready Flag RF	CHCR00.PRSEL=100 _B
	stf	Trigger Flag TF	CHCR00.PRSEL=101 _B
	eray_obusy	E-RAY Transfer Message RAM to Output Buffer RAM busy	CHCR00.PRSEL=110 _B
	eray_ibusy	E-RAY Transfer Output Buffer RAM to Message RAM busy	CHCR02.PRSEL=111 _B
01	ssc_tir	SSC Transmit Interrupt	CHCR01.PRSEL=000 _B
	ssc_rir	SSC Receive Interrupt	CHCR01.PRSEL=001 _B
	mli2_o	MLI	CHCR01.PRSEL=010 _B
	mli3_o	MLI	CHCR01.PRSEL=011 _B
	rt	Ready Flag RF	CHCR01.PRSEL=100 _B
	stf	Trigger Flag TF	CHCR01.PRSEL=101 _B
	eray_obusy	E-RAY Transfer Message RAM to Output Buffer RAM busy	CHCR01.PRSEL=110 _B
	eray_ibusy	E-RAY Transfer Output Buffer RAM to Message RAM busy	CHCR02.PRSEL=111 _B

Direct Memory Access Controller (DMA)

Table 5-4 DMA Request Assignment (cont'd)

DMA Channel	DMA Request Input	DMA Request Source	Selected by
02	ssc_tir	SSC Transmit Interrupt	CHCR02.PRSEL=000 _B
	ssc_rir	SSC Receive Interrupt	CHCR02.PRSEL=001 _B
	mli2_o	MLI	CHCR02.PRSEL=010 _B
	mli3_o	MLI	CHCR02.PRSEL=011 _B
	rt	Ready Flag RF	CHCR02.PRSEL=100 _B
	stf	Trigger Flag TF	CHCR02.PRSEL=101 _B
	eray_obusy	E-RAY Transfer Message RAM to Output Buffer RAM busy	CHCR02.PRSEL=110 _B
	eray_ibusy	E-RAY Transfer Output Buffer RAM to Message RAM busy	CHCR02.PRSEL=111 _B
03	ssc_tir	SSC Transmit Interrupt	CHCR03.PRSEL=000 _B
	ssc_rir	SSC Receive Interrupt	CHCR03.PRSEL=001 _B
	mli2_o	MLI	CHCR03.PRSEL=010 _B
	mli3_o	MLI	CHCR03.PRSEL=011 _B
	rt	Ready Flag RF	CHCR03.PRSEL=100 _B
	stf	Trigger Flag TF	CHCR03.PRSEL=101 _B
	eray_obusy	E-RAY Transfer Message RAM to Output Buffer RAM busy	CHCR02.PRSEL=110 _B
	eray_ibusy	E-RAY Transfer Output Buffer RAM to Message RAM busy	CHCR02.PRSEL=111 _B

Direct Memory Access Controller (DMA)

Table 5-4 DMA Request Assignment (cont'd)

DMA Channel	DMA Request Input	DMA Request Source	Selected by
04	ssc_tir	SSC Transmit Interrupt	CHCR04.PRSEL=000 _B
	ssc_rir	SSC Receive Interrupt	CHCR04.PRSEL=001 _B
	mli2_o	MLI	CHCR04.PRSEL=010 _B
	mli3_o	MLI	CHCR04.PRSEL=011 _B
	rt	Ready Flag RF	CHCR04.PRSEL=100 _B
	stf	Trigger Flag TF	CHCR04.PRSEL=101 _B
	eray_obusy	E-RAY Transfer Message RAM to Output Buffer RAM busy	CHCR04.PRSEL=110 _B
	eray_ibusy	E-RAY Transfer Output Buffer RAM to Message RAM busy	CHCR02.PRSEL=111 _B
05	ssc_tir	SSC Transmit Interrupt	CHCR05.PRSEL=000 _B
	ssc_rir	SSC Receive Interrupt	CHCR05.PRSEL=001 _B
	mli2_o	MLI	CHCR05.PRSEL=010 _B
	mli3_o	MLI	CHCR05.PRSEL=011 _B
	rt	Ready Flag RF	CHCR05.PRSEL=100 _B
	stf	Trigger Flag TF	CHCR05.PRSEL=101 _B
	eray_obusy	E-RAY Transfer Message RAM to Output Buffer RAM busy	CHCR05.PRSEL=110 _B
	eray_ibusy	E-RAY Transfer Output Buffer RAM to Message RAM busy	CHCR02.PRSEL=111 _B

Direct Memory Access Controller (DMA)

Table 5-4 DMA Request Assignment (cont'd)

DMA Channel	DMA Request Input	DMA Request Source	Selected by
06	ssc_tir	SSC Transmit Interrupt	CHCR06.PRSEL=000 _B
	ssc_rir	SSC Receive Interrupt	CHCR06.PRSEL=001 _B
	mli2_o	MLI	CHCR06.PRSEL=010 _B
	mli3_o	MLI	CHCR06.PRSEL=011 _B
	rt	Ready Flag RF	CHCR06.PRSEL=100 _B
	stf	Trigger Flag TF	CHCR06.PRSEL=101 _B
	eray_obusy	E-RAY Transfer Message RAM to Output Buffer RAM busy	CHCR02.PRSEL=110 _B
	eray_ibusy	E-RAY Transfer Output Buffer RAM to Message RAM busy	CHCR02.PRSEL=111 _B
07	ssc_tir	SSC Transmit Interrupt	CHCR07.PRSEL=000 _B
	ssc_rir	SSC Receive Interrupt	CHCR07.PRSEL=001 _B
	mli2_o	MLI	CHCR07.PRSEL=010 _B
	mli3_o	MLI	CHCR07.PRSEL=011 _B
	rt	Ready Flag RF	CHCR07.PRSEL=100 _B
	stf	Trigger Flag TF	CHCR07.PRSEL=101 _B
	eray_obusy	E-RAY Transfer Message RAM to Output Buffer RAM busy	CHCR02.PRSEL=110 _B
	eray_ibusy	E-RAY Transfer Output Buffer RAM to Message RAM busy	CHCR02.PRSEL=111 _B

5.3.2 Address Map

In the SAK-CIC310-OSMX2HT, the registers of the DMA module are located in the following address range:

- Module Base Address = 0000 0400_H
Module End Address = 0000 05FF_H
- Absolute Register Address = Module Base Address + Offset Address
(offset addresses see [Table 5-2](#))

6 Synchronous Serial Interface (SSC)

This chapter describes the SSC synchronous serial interface of the SAK-CIC310-OSMX2HT. It contains the following sections:

- Functional description of the SSC Kernel
(see [Section 6.1](#))
- SSC kernel register description, describes all SSC Kernel specific registers
(see [Section 6.2](#))

6.1 SSC Kernel Description

Figure 6-1 shows a global view of all functional blocks of the SSC interface.

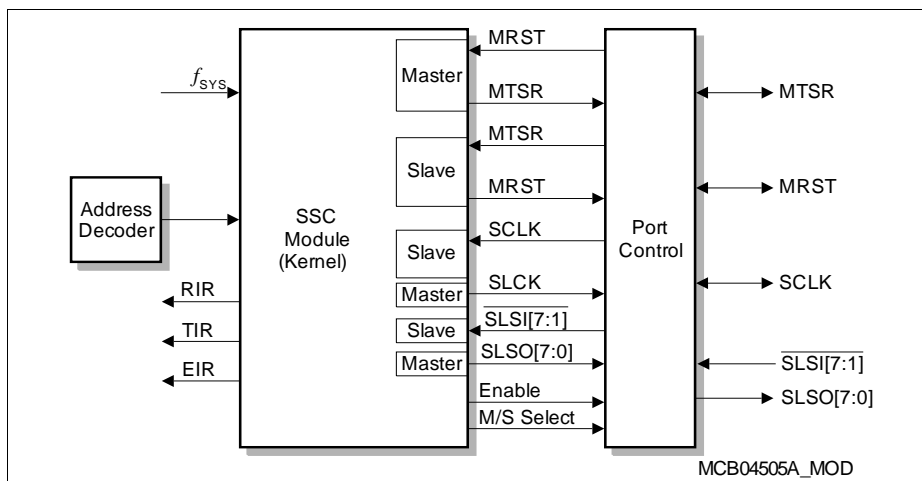


Figure 6-1 General Block Diagram of the SSC Interface

6.1.1 Overview

The SSC supports full-duplex and half-duplex serial synchronous communication up to 40 Mbaud (assuming 80 MHz module clock). The serial clock signal can be generated by the SSC itself (master mode) or can be received from an external master (slave mode). Data width, shift direction, clock polarity and phase are programmable. This allows communication with SPI-compatible devices. Transmission and reception of data is double-buffered. A shift clock generator provides the SSC with a separate serial clock signal. Eight slave select inputs are available for slave mode operation. Eight programmable slave select outputs (chip selects) are supported in master mode.

Synchronous Serial Interface (SSC)**Features**

- Master and slave mode operation
 - Full-duplex or half-duplex operation
 - Automatic pad control possible
- Flexible data format
 - Programmable number of data bits: 2 to 16 bit
 - Programmable shift direction: LSB or MSB shift first
 - Programmable clock polarity: idle low or high state for the shift clock
 - Programmable clock/data phase: data shift with leading or trailing edge of the shift clock
- Baud rate generation from 40 Mbaud to 610,3 baud (assuming 80 MHz module clock)
- Interrupt generation
 - On a transmitter empty condition
 - On a receiver full condition
 - On an error condition (receive, phase, baud rate, transmit error)
- Four-pin interface
- Flexible SSC pin configuration
- Up to eight slave select inputs in slave mode
- Up to eight programmable slave select outputs SLSO in master mode
 - Automatic SLSO generation with programmable timing
 - Programmable active level and enable control

6.1.2 General Operation

The SSC supports full-duplex and half-duplex synchronous communication up to 40 Mbaud (assuming 80 MHz module clock). The serial clock signal can be generated by the SSC itself (master mode) or be received from an external master (slave mode). Data width, shift direction, clock polarity and phase are programmable. This allows communication with SPI-compatible devices. Transmission and reception of data are double-buffered. A shift clock generator provides the SSC with a separate serial clock signal.

Configuration of the high-speed synchronous serial interface is very flexible, so it can work with other synchronous serial interfaces, can serve for master/slave or multimaster interconnections, or can operate compatibly with the popular SPI interface. It can be used to communicate with shift registers (I/O expansion), peripherals (e.g. EEPROMs etc.), or other controllers (networking). The SSC supports half-duplex and full-duplex communication. Data is transmitted or received on pins MTSR (Master Transmit / Slave Receive) and MRST (Master Receive/Slave Transmit). The clock signal is output or input via pin SCLK (Serial Clock). These three pins are typically alternate output functions of port pins. If they are implemented as dedicated bi-directional pins they can be directly controlled by the SSC. In slave mode the SSC can be selected from a master via

Synchronous Serial Interface (SSC)

dedicated slave select input lines (SLSI). In master mode automatic generation of slave select output lines (SLSO) is supported.

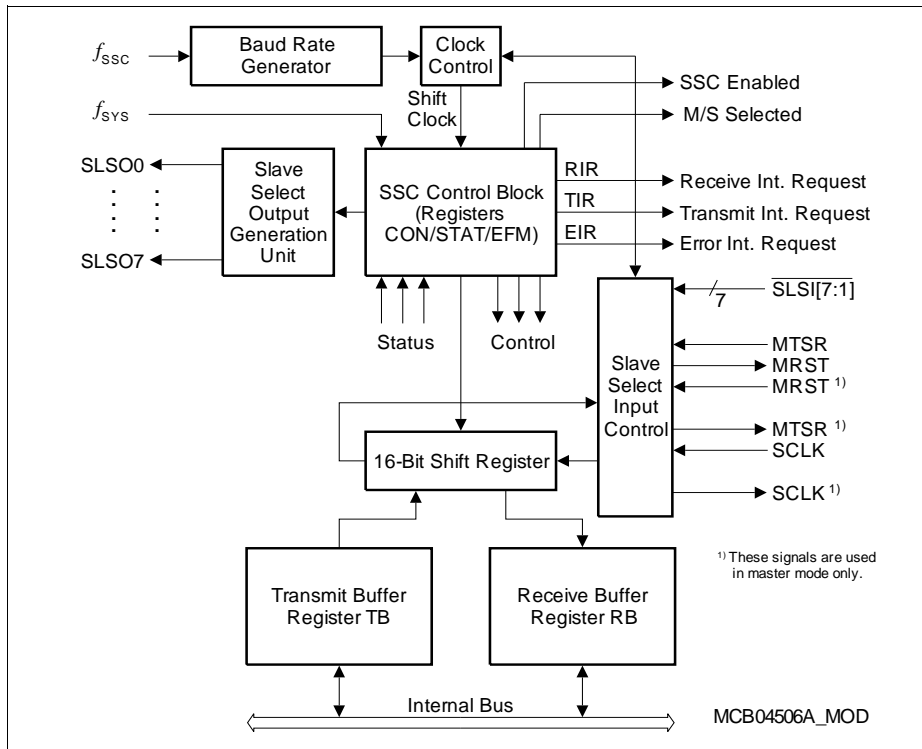


Figure 6-2 Synchronous Serial Channel SSC Block Diagram

6.1.2.1 Operating Mode Selection

The operating mode of the serial channel SSC is controlled by its control register, CON. Status information is contained in its status register, STAT.

The shift register of the SSC is connected to both the transmit pin and the receive pin via the pin control logic (see block diagram in [Figure 6-2](#)). Transmission and reception of serial data are synchronized and take place at the same time, that is, the same number of transmitted bits is also received. Transmit data is written into the Transmit Buffer TB. It is moved to the shift register as soon as this is empty. An SSC master (CON.MS = 1) immediately begins transmitting, while an SSC slave (CON.MS = 0) will wait for an active shift clock. When the transfer starts, the busy flag STAT.BSY is set and the Transmit Interrupt Request line (TIR) will be activated to indicate that register Transmit Buffer (TB)

Synchronous Serial Interface (SSC)

may be reloaded. When the number of bits (2 to 16, as programmed) have been transferred, the contents of the shift register are moved to the Receive Buffer (RB) and the Receive Interrupt Request line (RIR) will be activated. If no further transfer is to take place (TB is empty), STAT.BSY will be cleared at the same time. Software should not modify STAT.BSY, as this flag is hardware controlled.

Note: Only one SSC (etc.) can be master at a given time.

The transfer of serial data bits can be programmed in many respects:

- The data width can be selected from 2 bits to 16 bits
- A transfer may start with the LSB or the MSB
- The shift clock may be idle low or idle high
- The data bits may be shifted with the leading or trailing edge of the clock signal
- The baud rate (shift clock) can be set from 610,3 baud up to 40 Mbaud (assuming 80 MHz module clock)
- The shift clock can be generated (master) or received (slave)

These features allow the SSC to be adapted to a wide range of applications that require serial data transfer.

The Data Width Selection supports the transfer of frames of any data length from 2-bit “characters” up to 16-bit “characters”. Starting with the LSB (CON.HB = 0) allows communication with such devices as an SSC device in synchronous mode or 8051-like serial interfaces. Starting with the MSB (CON.HB = 1) allows operation compatible with the SPI interface.

Regardless of the data width selected and whether the MSB or the LSB is transmitted first, the transfer data is always right-aligned in registers TB and RB, with the LSB of the transfer data in bit 0 of these registers. The data bits are rearranged for transfer by the internal shift register logic. The unselected bits of TB are ignored, the unselected bits of RB will not be valid and should be ignored by the receiver service routine.

The Clock Control allows the adaptation of transmit and receive behavior of the SSC to a variety of serial interfaces. A specific clock edge (rising or falling) is used to shift out transmit data, while the other clock edge is used to latch in receive data. Bit CON.PH selects the leading edge or the trailing edge for each function. Bit CON.PO selects the level of the clock line in the idle state. So for an idle-high clock, the leading edge is a falling one, a 1-to-0 transition (see [Figure 6-3](#)).

*Note: CON.PH=1 may not be configured in slave mode, because the first bit to be transmitted (LSB or MSB of the transmit buffer TB) on the MRST line after the falling edge of the SLSIx is replaced by the **PISEL.STIP** value. In master mode in most cases the first received bit is invalid.*

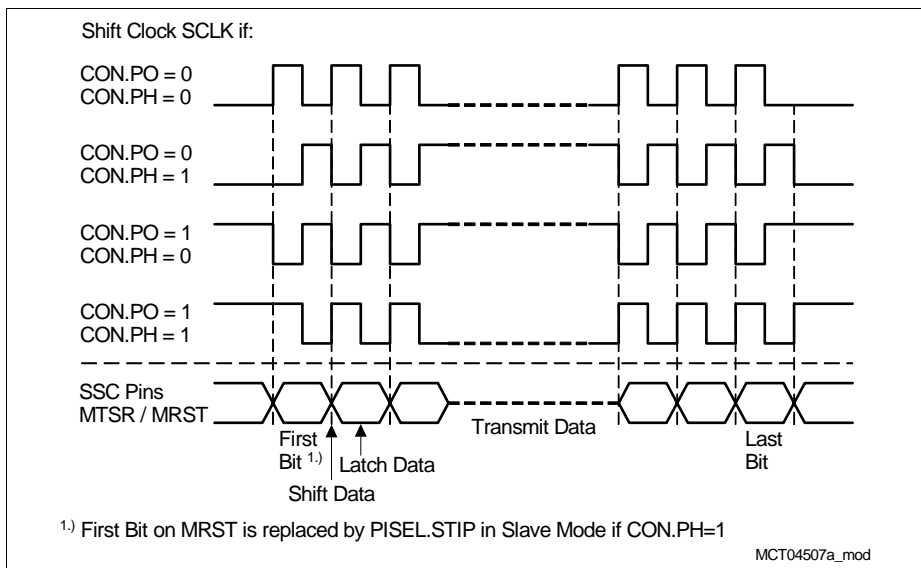


Figure 6-3 Serial Clock Phase and Polarity Options (Master Mode)

6.1.2.2 Full-Duplex Operation

Note: The description in this section assumes that the SSC is used with software controlled bi-directional GPIO port lines that provide open-drain capability (see also [Section 6.1.2.5](#)).

The various devices are connected through three lines. The definition of these lines is always determined by the master. The line connected to the master's data output pin MTSR is the transmit line, the receive line is connected to its data input line MRST, and the clock line is connected to pin SCLK. Only the device selected for master operation generates and outputs the serial clock on pin SCLK. All slaves receive this clock, so their pin SCLK must be switched to input mode. The output of the master's shift register is connected to the external transmit line, which in turn is connected to the slaves' shift register input. The output of the slaves' shift register is connected to the external receive line in order to enable the master to receive the data shifted out of the slave. The external connections are hard-wired, with the function and direction of these pins determined by the master or slave operation of the individual device.

Note: The shift direction shown in [Figure 6-4](#) applies to both MSB first and LSB first operation.

Synchronous Serial Interface (SSC)

When initializing the devices in this configuration, one device must be selected for master operation while all other devices must be programmed for slave operation. Initialization includes the operating mode of the device's SSC and also the function of the respective port lines.

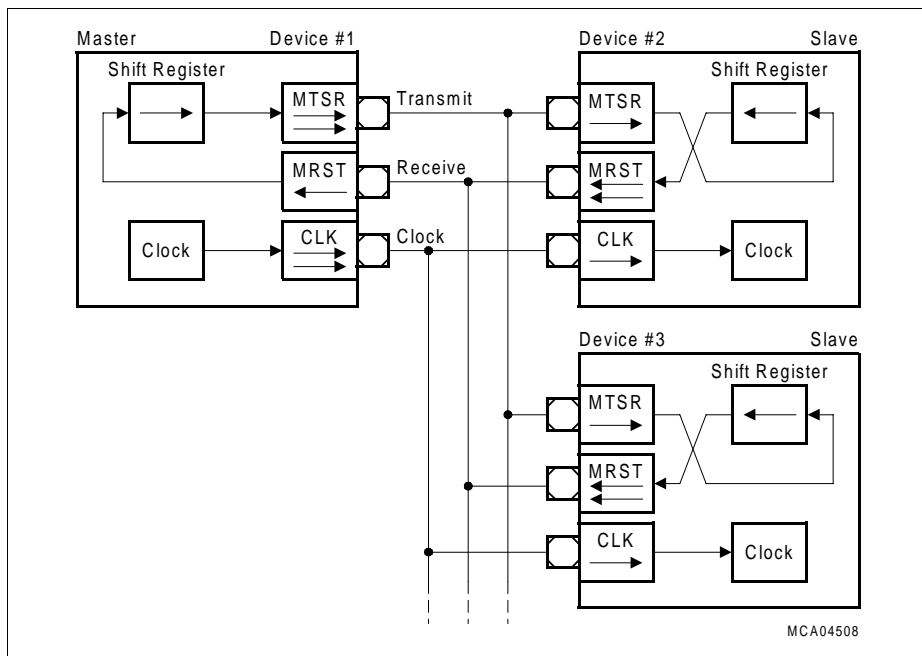


Figure 6-4 SSC Full-Duplex Configuration

The data output pins MRST of all slave devices are connected onto one receive line in this configuration. During a transfer each slave shifts out data from its shift register. There are two ways to avoid collisions on the receive line due to different slave data:

- **Only one slave drives the line** and enables the driver of its MRST pin. All the other slaves must program their MRST pins to input. So, only one slave can put its data onto the master's receive line. Only reception of data from the master is possible. The master selects the slave device from which it expects data either by separate select lines, or by sending a special command to this slave. The selected slave then switches its MRST line to output until it gets a de-selection signal or command.
- **The slaves use open drain output on MRST.** This forms a wired-AND connection. The receive line needs an external pull-up in this case. Corruption of the data on the receive line sent by the selected slave is avoided when all slaves not selected for transmission to the master send only 1s. Since this high level is not actively driven

Synchronous Serial Interface (SSC)

onto the line, but is only held through the pull-up device, the selected slave can pull this line actively to a low-level when transmitting a zero bit. The master selects the slave device from which it expects data either by separate select lines, or by sending a special command to this slave.

After performing all necessary initializations of the SSC, the serial interfaces can be enabled. For a master device, the alternate clock line will now go to its programmed polarity. The alternate data line will go to either 0 or 1, until the first transfer will start. After a transfer, the alternate data line will always remain at the logic level of the last transmitted data bit.

When the serial interfaces are enabled, the master device can initiate the first data transfer by writing the transmit data into register TB. This value is copied into the shift register (assumed to be empty at this time), and the selected first bit of the transmit data will be placed onto the MTSR line on the next clock from the shift clock generator (transmission only starts, if CON.EN = 1). Depending on the selected clock phase, also a clock pulse will be generated on the SCLK line. With the opposite clock edge, the master simultaneously latches and shifts in the data detected at its input line MRST. This “exchanges” the transmit data with the receive data. Because the clock line is connected to all slaves, their shift registers will be shifted synchronously with the master’s shift register, shifting out the data contained in the registers, and shifting in the data detected at the input line. After the pre-programmed number of clock pulses (via the data width selection), the data transmitted by the master is contained in all slaves’ shift registers, while the master’s shift register holds the data of the selected slave. In the master and all slaves, the content of the shift register is copied into the Receive Buffer (RB) and the Receive Interrupt Line (RIR) is activated.

A slave device will immediately output the selected first bit (MSB or LSB of the transfer data) at pin MRST when the contents of the transmit buffer are copied into the slave’s shift register. Bit STAT.BSY is not set until the first clock edge at SCLK appears. The slave device will not wait for the next clock from the shift clock generator — as the master does — because the first clock edge generated by the master may be already used to clock in the first data bit, depending on the selected clock phase. So the slave’s first data bit must already be valid at this time.

Note: On the SSC a transmission and a reception always takes place at the same time, regardless whether valid data has been transmitted or received.

6.1.2.3 Half-Duplex Operation

Note: The description in this section assumes that the SSC is used with software controlled bi-directional GPIO port lines that provide open-drain capability (see also [Section 6.1.2.5](#)).

In a half-duplex configuration, only one data line is necessary for both receiving and transmitting data. The data exchange line is connected to one pin of the SAK-CIC310-OSMX2HT, the MTSR, the clock line is connected to the SCLK pin.

Synchronous Serial Interface (SSC)

The master device controls the data transfer by generating the shift clock, while the slave devices receive it. Further more the master device controls the data transfer direction by generating the direction (DIR) signal, while the slave devices receive it and switches respectively the MTSR pin as MTSR input or MTST output. Due to the fact that all transmit and receive pins are connected to the one data exchange line, serial data may be moved between arbitrary stations.

Similar to full-duplex mode there are **three ways to avoid collisions** on the data exchange line:

- Only the transmitting device may enable its transmit pin driver
- The non-transmitting devices use open drain output and only send 1's
- An external direction signal switches the pin driver between input and output characteristics.

Because the data inputs and outputs are connected together, a transmitting device will clock in its own data at the input pin (MRST for a master device, MTSR for a slave). In this way, any corruption is detected on the common data exchange line where the received data is not equal to the transmitted data.

The master enables one of the connected slaves with the associated direction select input signal DIR. The output signal of the slaves (MRST), which is a input of the master, is activated by the DIR signal and the input signal of the slave (MTSR), which is a output of the master, is deactivated by all other slaves. The user has to take care that the host controller respectively drives the single line. The switching between the output (MRST) and input (MTSR) of the deselected slaves is accomplished by deactivating the output stage on the associated pads. This deactivation takes place, if MODE = 11 and DIR = 1, i.e. SSC selected and direction selected.

IOCRB = SSC_EERCUM.MSCON and DIR and MODE=11

An additional bit field PC6B in register P1_IOC4 for port P1.6 is defined with the same definition PC6 as in [Table 9-2](#). If the upper condition is true, the signals OCRB6 become true, then the new bit fields PC6B defines the GPIO port behavior, otherwise the regular PC6 defines the GPIO port behavior. This new bit field is programmable and could be programmed to an input-function in this case. The port thus becomes input, whenever DIR goes high (and respectively PC6 is programmed as output, so the port becomes output whenever DIR goes low).

Synchronous Serial Interface (SSC)

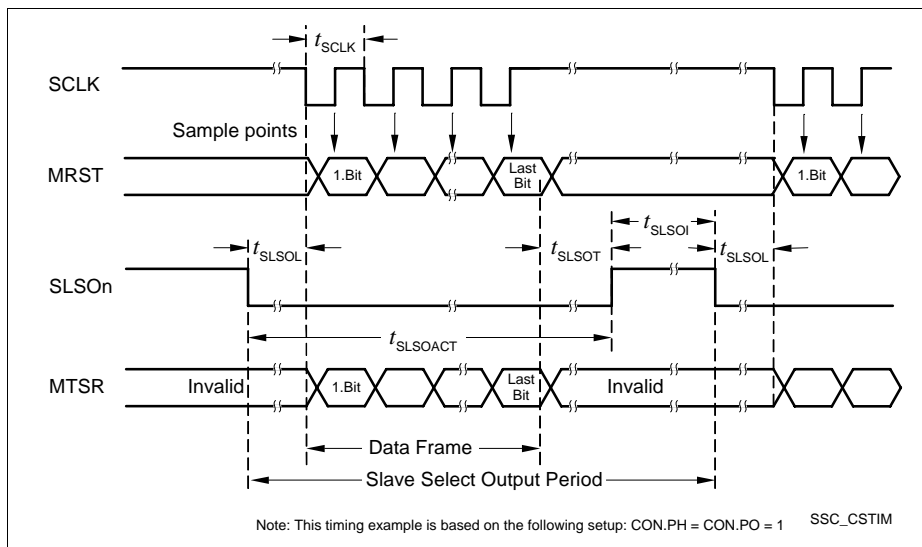


Figure 6-5 SSC Half-Duplex Configuration

6.1.2.4 Continuous Transfers

When the transmit interrupt request flag is set, it indicates that the Transmit Buffer (TB) is empty and is ready to be loaded with the next transmit data. If TB has been reloaded by the time the current transmission of a frame is finished, the data is immediately transferred to the shift register and the next transmission of a frame may start without any additional delay (if selected respectively in the bitfields of the **SSOTC** register). On the data line there is no gap between the two successive frames if respectively no delays are selected within the SLSON register. For example, two byte transfers would look the same as one word transfer. This feature can be used to interface with devices that can operate with or require more than 16 data bits per transfer. It is just a matter for software how long a total data length can be. This option can also be used e.g. to interface to byte-wide and word-wide devices on the same serial bus.

Note: Of course, this can only happen in multiples of the selected basic data width, because it would require disabling/enabling of the SSC to reprogram the basic data width on-the-fly.

6.1.2.5 Port Control

The SSC uses three lines to communicate with the external world. Pin SCLK serves as the clock line, while pins MRST (Master Receive/Slave Transmit) and MTSR (Master

Synchronous Serial Interface (SSC)

Transmit/Slave Receive) serve as the serial data input/output lines. As shown in [Figure 6-1](#) these three lines (SCLK as input, Master Receive, Slave Receive) have all two inputs at the SSC Module kernel. Three bits in register PISEL define which of the two kernel inputs (A or B) are connected. This feature allows for each of the three SSC communication lines to be connected to two inputs.

Operation of the SSC I/O lines depends on the selected operating mode (master or slave). The direction of the port lines depends on the operating mode. The SSC will automatically use the correct kernel output or kernel input line of the ports when switching modes. Port pins assigned as SSC I/O lines can be controlled in two ways:

- By hardware
- By software

When the SSC I/O lines are connected with dedicated pins typically hardware I/O control should be used. In this case, two output signals reflect directly the state of the CON.EN and CON.MS bits (the M/S select line is inverted to the CON.MS bit definition).

When the SSC I/O lines are connected with bi-directional lines of general purpose I/O ports typically software I/O control should be used. In this case port registers must be programmed for alternate output and input selection. When switching between master and slave mode port registers must be reprogrammed.

Using the open-drain output feature of port lines helps avoid bus contention problems and reduces the need for hard-wired hand-shaking or slave select lines. In this case, it is not always necessary to switch the direction of a port pin. Note that in hardware controlled I/O mode the availability of open-drain outputs depends on the type of the used dedicated output pins. The SSC module itself does not provide any control capability for open drain control.

6.1.2.6 Baud Rate Generation

The serial channel SSC has its own dedicated 16-bit baud rate generator with 16-bit reload capability, allowing baud rate generation independent from the timers. In addition to [Figure 6-2](#), [Figure 6-6](#) shows the baud rate generator of the SSC in more detail.

Synchronous Serial Interface (SSC)

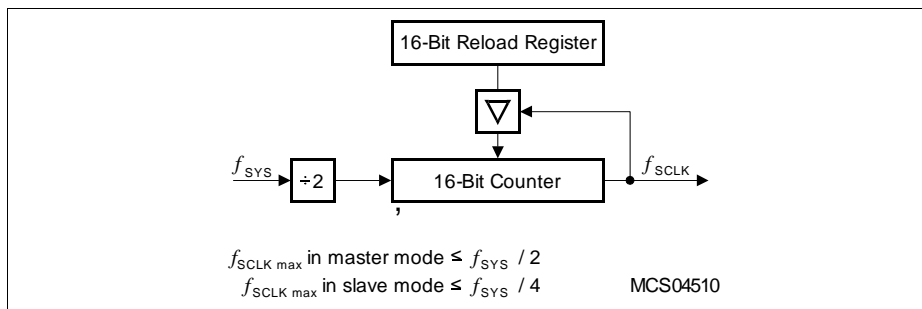


Figure 6-6 SSC Baud Rate Generator

The baud rate generator is clocked with the module clock f_{SYS} . The timer counts downwards. Register BR is the dual-function Baud Rate Generator/Reload register. Reading BR, while the SSC is enabled, returns the contents of the timer. Reading BR, while the SSC is disabled, returns the programmed reload value. In this mode, the desired reload value can be written to BR.

Note: Never write to BR while the SSC is enabled.

The formulas below calculate either the resulting baud rate for a given reload value, or the required reload value for a given baud rate:

$$\text{Baud Rate}_{\text{SSC}} = \frac{f_{\text{SYS}}}{2 \times (\text{BR_VALUE} + 1)} \quad (6.1)$$

$$\text{BR_VALUE} = \frac{f_{\text{SYS}}}{2 \times \text{Baud Rate}_{\text{SSC}}} - 1 \quad (6.2)$$

BR_VALUE represents the content of the reload register, taken as unsigned 16-bit integer while $\text{Baud_Rate}_{\text{SSC}}$ is equal to f_{SCLK} as shown in [Figure 6-6](#).

The maximum baud rate that can be achieved when using a module clock of 80 MHz is 40 Mbaud in master mode (with $\langle \text{BR} \rangle = 0000_{\text{H}}$) and 20 Mbaud in slave mode (with $\langle \text{BR} \rangle = 0001_{\text{H}}$).

Table 6-1 lists some possible baud rates together with the required reload values and the resulting bit times, assuming a module clock of 80 MHz.

Table 6-1 Typical Baud Rates of the SSC ($f_{\text{SYS}} = 80 \text{ MHz}$)

Reload Value	Baud Rate ($= f_{\text{SCLK}}$)	Deviation
0000 _H	40.00 Mbaud (only in master mode)	0.0%
0001 _H	20.00 Mbaud	0.0%
0002 _H	13.333 Mbaud	0.0%
0003 _H	10.00 Mbaud	0.0%
0027 _H	1 Mbaud	0.0%
018F _H	100 kbaud	0.0%
0F9F _H	10 kbaud	0.0%
9C3F _H	1 kbaud	0.0%
FFFF _H	610.352 baud	0.0%

6.1.2.7 Slave Select Input Operation

For systems with multiple slaves, the SSC module provides $\overline{\text{SLSI}}$ slave select input lines, that allows to enable/disable the SCLK, MTSR, and MRST signals in slave mode. Slave mode is selected by $\text{CON.MS} = 0$. The $\overline{\text{SLSI}}$ input logic shown in [Figure 6-7](#) is controlled by register PISEL and CON.

With $\text{PISEL.SLSIS} = 0$ and slave mode selected, the $\overline{\text{SLSI}}[7:1]$ lines do not control the SSC I/O lines. The slave receive input signal at pins MTSRA or MTSRB and the slave clock signal at pin SCLKA or SCLKB are passed further to MTSRI and SCLKI. The slave transmit signal MRSTI is passed directly to MRST.

With $\text{PISEL.SLSIS} = 1$, slave select mode is enabled and input signals $\overline{\text{SLSI}}[7:1]$ control the operation of the SSC I/O lines as follows:

- $\overline{\text{SLSIx}} = 1$: SSC slave is not selected.
 - MTSRI is connected with the slave receive input signals MTSRA or MTSRB, depending on PISEL.SRIS (slave mode receive input select).
 - MRST is driven with the logic level of bit PISEL.STIP (slave transmit idle state).
 - SCLKI is driven with the logic level of CON.PO (clock polarity control).
- $\overline{\text{SLSIx}} = 0$: SSC is selected as slave.
 - MTSRI is connected with the slave receive input signals MTSRA or MTSRB, depending on PISEL.SRIS (slave mode receive input select).
 - MRST is directly driven with the slave transmit output signal MRSTI.
 - SCLKI is connected with the slave clock input signals SCLKA or SCLKB, depending on PISEL.SCIS (slave mode clock input select).

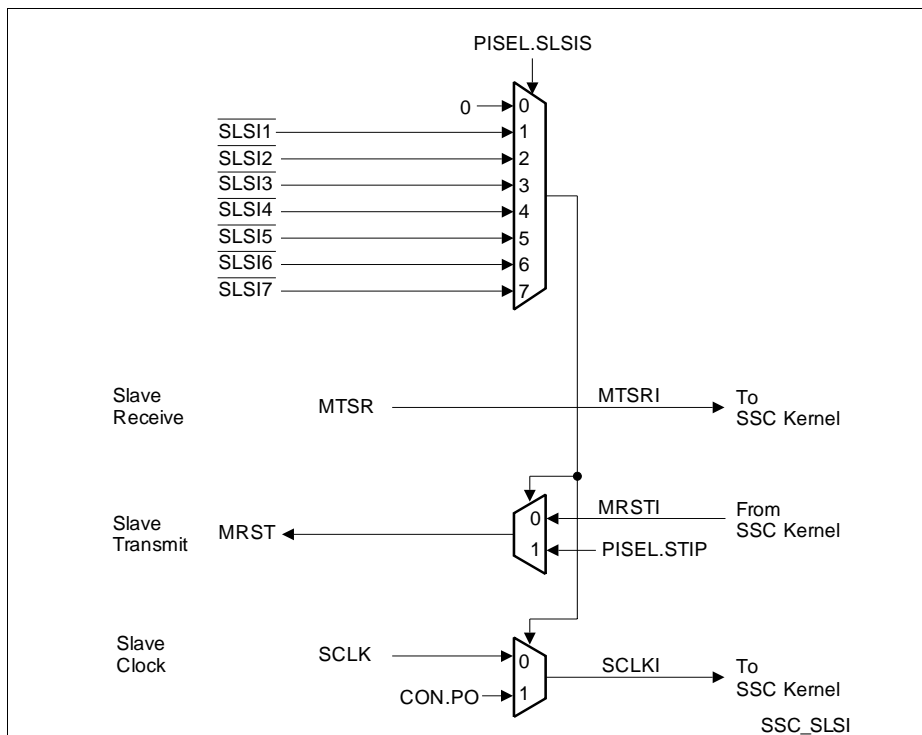


Figure 6-7 Slave Select Input Logic

6.1.2.8 Slave Select Output Generation Unit

In Master Mode, the slave select output generation unit of the SSC automatically generates up to eight slave select output lines SLSO[7:0] for serial transmit operations. The slave select output generation unit further allows to adjust the chip select timing parameters. The active/inactive state of a slave select output as well as the enable/disable state can be controlled individually for each slave select output (see [Figure 6-9](#)). The basic slave select output timing is shown in [Figure 6-8](#), assuming an low active level of the SLSOn lines.

Synchronous Serial Interface (SSC)

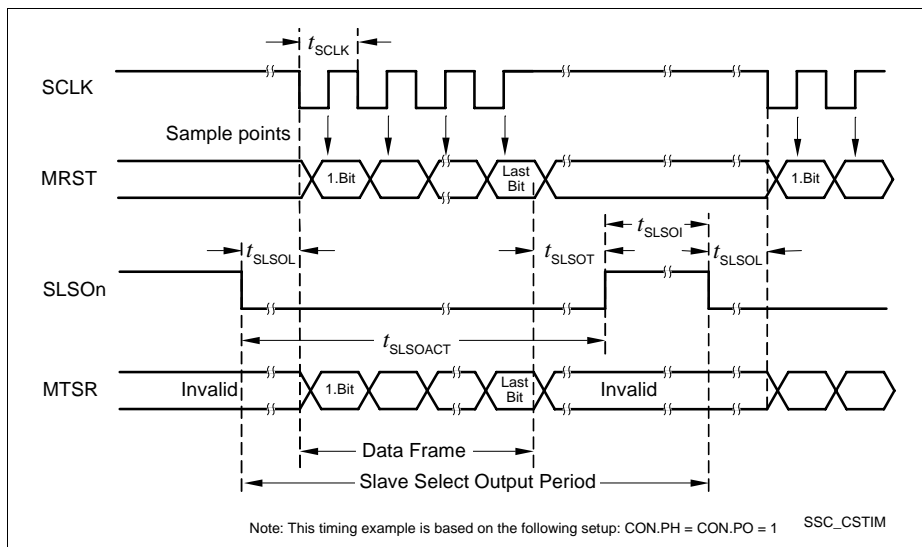


Figure 6-8 SSC Slave Select Output Timing

A slave select output period always starts after a serial write operation to register TB. Afterwards SLSON becomes active (low) for a number of SCLK cycles (leading delay cycles) before the first bit of the serial data stream occurs at MTSR. After the transmission of the data frame SLSOx remains active (low) for a number of SCLK cycles (trailing delay cycles) before it becomes again inactive. This inactive state of SLSON is valid at least for a number of SCLK cycles (inactive delay cycles) before a new chip select period can be started.

The three parameters of a chip select period are controlled by bit fields in the slave select output timing control register SSOTC. Each of these bit fields can contain a value from 0 to 3 defining delay cycles of 0 to 3 multiples of the t_{SCLK} shift clock period. The three parameters are:

- Number of leading delay cycles ($t_{\text{SLSOL}} = \text{SSOTC.LEAD} \times t_{\text{SCLK}}$)
- Number of trailing delay cycles ($t_{\text{SLSOT}} = \text{SSOTC.TRAIL} \times t_{\text{SCLK}}$)
- Number of inactive delay cycles ($t_{\text{SLSOI}} = \text{SSOTC.INACT} \times t_{\text{SCLK}}$)

If SSOTC.INACT = 00_b and register TB has already been loaded with the data for the next data frame, the next chip select period is started with its leading delay phase without SLSON going inactive. If, in this case, TB has not been loaded in time with the data for the next data frame, SLSOx becomes inactive again.

Synchronous Serial Interface (SSC)

Slave Select Output Control

Each slave select output SLSON can be enabled individually. When SSOC.OENn = 0, SLSON is enabled. Further, active and inactive levels of the SLSON outputs are programmable. Bit SSOC.AOLn defines the state of the active level of SLSON.

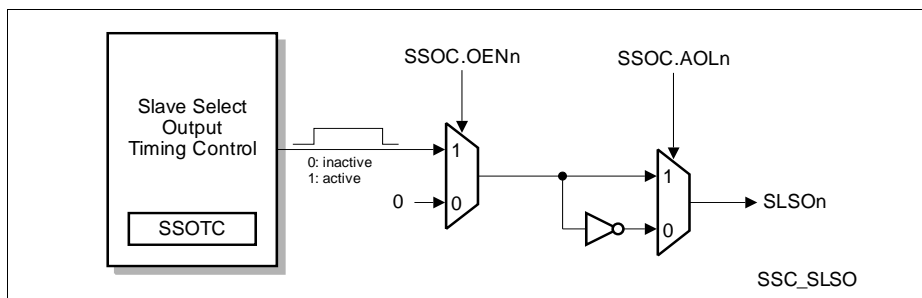


Figure 6-9 Slave Select Output Control Logic

Slave Select Output 7 Delayed Mode

In the SLSO7 delayed mode (SSOTC.SLSO7MOD = 1), the timing of the slave select output SLSO7 as programmed by the three parameters in SSOTC (number of trailing, leading, and inactive delay clock cycles) is delayed by one shift clock period for the inactive to active edge. The active to inactive edge is not delayed. The timing of SLSO7 in the delayed mode is shown in [Figure 6-10](#). The bold lines show the timing of SLSO7 in normal operating mode and the dotted lines show the timing of SLSO7 in delayed mode.

Synchronous Serial Interface (SSC)

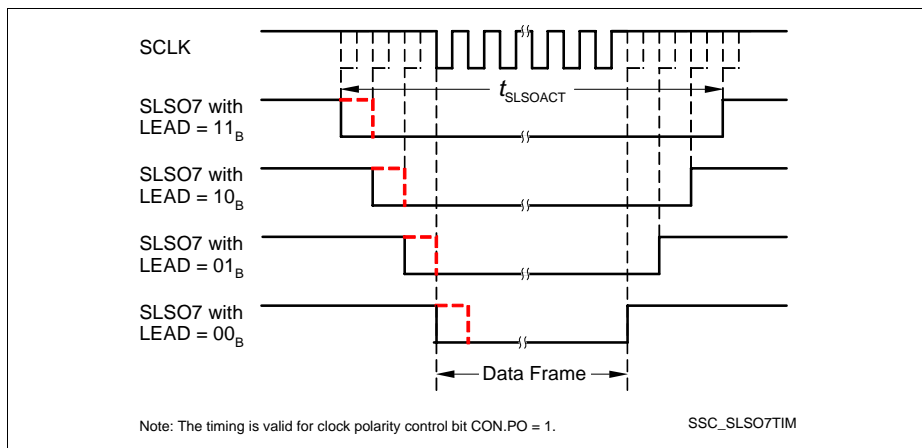


Figure 6-10 SLSO7 Delayed Mode

Slave Select Register Update

The bits in the registers SSOC and SSOTC are buffered while a transfer is in progress. The buffer samples the values written to these registers in the following case:

- Start of the internal transfer sequence

So it's always guaranteed that the data of one SSC transfer is transmitted with one constant slave select configuration and a configuration change is only valid with the start of the next new SSC transfer.

6.1.2.9 Shift Clock Generation

The serial channel SSC operates with its own shift clock f_{SYS} .

6.1.2.10 Error Detection Mechanisms

The SSC is able to detect four different error conditions. Receive Error and Phase Error are detected in all modes, while Transmit Error and Baud Rate Error apply to Slave Mode only. In case of a Transmit Error or Receive Error, the respective error flags are always set and the error interrupt requests will be generated by activating the EIR line only if the corresponding error enable bits have been set (see [Figure 6-11](#)). The error interrupt handler may then check the error flags to determine the cause of the error interrupt. The error flags are not reset automatically, but must be cleared via register EFM after servicing. This allows servicing of some error conditions via interrupt, while others may be polled by software. The error status flags can be set and reset by software via the error flag modification register EFM.

Synchronous Serial Interface (SSC)

Note: The error interrupt handler must clear the associated (enabled) error flag(s) to prevent repeated interrupt requests.

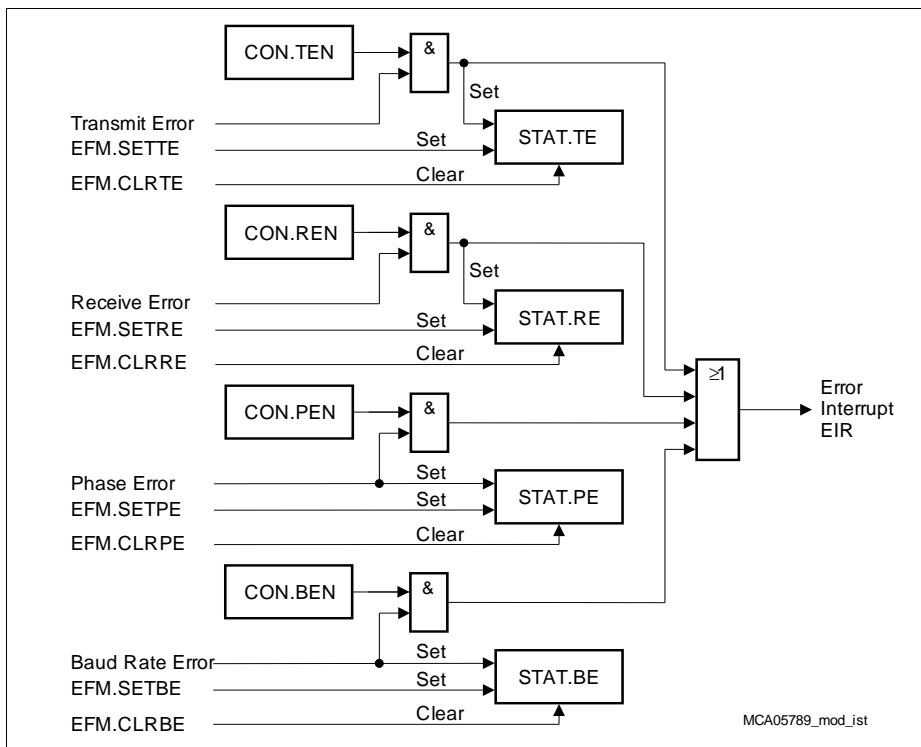


Figure 6-11 SSC Error Interrupt Control

A **Receive Error** (Master or Slave mode) is detected when a new data frame is completely received, but the previous data was not read out of the receive buffer register RB. If enabled via CON.REN, this condition sets the error flag STAT.RE and activates the error interrupt request line EIR. This condition sets the error flag STAT.RE and, if enabled via CON.REN, sets the error interrupt request line EIR. The old data in the receive buffer RB will be overwritten with the new value and is irretrievably lost.

A **Phase Error** (Master or Slave Mode) is detected when the incoming data at pin MRST (Master Mode) or MTSR (Slave Mode), sampled with the same frequency as the module clock, changes between one cycle before and two cycles after the latching edge of the shift clock signal SCLK. This condition sets the error flag CON.PE and, when enabled via CON.PEN, the error interrupt request line EIR.

Synchronous Serial Interface (SSC)

Note: When CON.PH = 1, the data output signal may be disturbed shortly when the slave select input signal is changed after a serial transmission, resulting in a phase error.

A **Baud Rate Error** (Slave mode) is detected when the incoming clock signal deviates from the programmed baud rate (shift clock) by more than 100%, meaning it is either more than double or less than half the expected baud rate. This condition sets the error status flag STAT.BE and, if enabled by CON.BEN, the error interrupt request line EIR. Using this error detection capability requires that the slave's shift clock generator is programmed to the same baud rate as the master device. This feature detects false additional pulses or missing pulses on the clock line (within a certain frame).

Note: If this error condition occurs and bit CON.AREN = 1, an automatic reset of the SSC will be performed. This is done to re-initialize the SSC, if too few or too many clock pulses have been detected.

Note: This error can occur after any transfer if the communication is stopped. This is the case due to the fact that SSC module supports back-to-back transfers for multiple transfers. In order to handle this the baud rate detection logic expects after a finished transfer immediately a next clock cycle for a new transfer.

A **Transmit Error** (Slave Mode) is detected when a transfer was initiated by the master (shift clock gets active), but the transmit buffer (TB) of the slave was not updated since the last transfer. If enabled via CON.TEN, this condition sets the error status flag STAT.TE and activates the EIR line. This condition sets the error status flag STAT.TE and, if enabled via CON.TEN, the EIR line. If a transfer starts while the transmit buffer is not updated, the slave will shift out the 'old' contents of the shift register, which is normally the data received during the last transfer. This may lead to the corruption of the data on the transmit/receive line in Half-duplex Mode (open drain configuration) if this slave is not selected for transmission. This mode requires that slaves not selected for transmission only shift out ones, thus, their transmit buffers must be loaded with FFFF_H prior to any transfer.

Note: A slave with push/pull output drivers not selected for transmission, will normally have its output drivers switched. However, to avoid possible conflicts or misinterpretations, it is recommended to always load the slave's transmit buffer prior to any transfer.

The cause of an error interrupt request (receive, phase, baud rate, transmit error) can be identified by the error status flags in control register CON.

Note: In contrast to the EIR line, the error status flags STAT.TE, STAT.RE, STAT.PE, and STAT.BE, are not cleared automatically upon entry into the error interrupt service routine, but must be cleared by software.

Synchronous Serial Interface (SSC)

6.2 SSC Kernel Registers

Figure 6-12 shows all registers associated with the SSC Kernel.

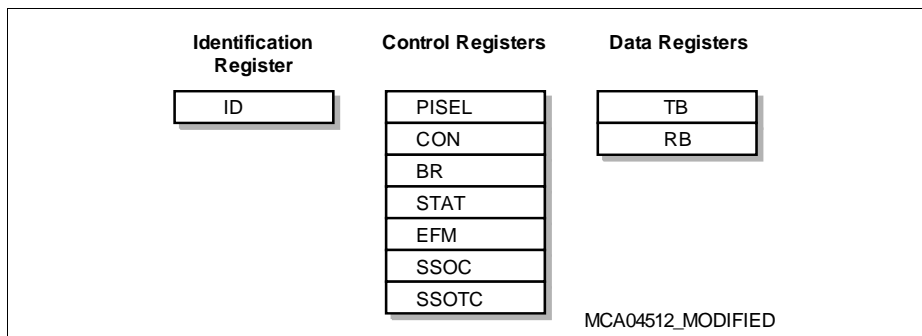


Figure 6-12 SSC Kernel Registers

Table 6-2 SSC Kernel Register Address Space

Module	Base Address	End Address	Note
SSC	0000 0900 _H	0000 09FF _H	256 byte

Table 6-3 SSC Kernel Registers

Register Short Name	Register Long Name	Offset Address	Description see
PISEL	Port Input Select Register	0004 _H	Page 6-20
ID	Module Identification Register	0008 _H	Page 6-22
	Reserved Register	0008 _H	-
CON	Control Register	0010 _H	Page 6-23
BR	Baud Rate Timer Reload Register	0014 _H	Page 6-32
SSOC	Slave Select Output Control Register	0018 _H	Page 6-29
SSOTC	Slave Select Output Timing Control Register	001C _H	Page 6-30
TB	Transmit Buffer Register	0020 _H	Page 6-33
RB	Receive Buffer Register	0024 _H	Page 6-34
STAT	Status Register	0028 _H	Page 6-25
EFM	Error Flag Modification Register	002C _H	Page 6-27

Synchronous Serial Interface (SSC)

The PISEL register controls the input signal selection of the SSC module. Each input of the module kernel receive, transmit and clock signals has associated two input lines (port A and port B).

PISEL

Port Input Select Register

(04_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0							STIP	0	SLSIS		SCIS	SRIS	MRIS		
r							rw	r	rw		rw	rw	rw	rw	

Field	Bits	Type	Description
MRIS	0	rw	Master Mode Receive Input Select MRIS selects the receive input line in master mode. 0 Receive input line MRSTA is selected 1 Receive input line MRSTB is selected
SRIS	1	rw	Slave Mode Receive Input Select SRIS selects receive input line that in slave mode. 0 Receive input line MTSRA is selected 1 Receive input line MTSRB is selected
SCIS	2	rw	Slave Mode Clock Input Select SCIS selects the module kernel SCLK input line that is used as clock input line in slave mode. 0 Slave mode clock input line SCLKA is selected 1 Slave mode clock input line SCLKB is selected

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
SLSIS	[5:3]	rw	Slave Mode Slave Select Input Selection 000 _B Slave select input lines are deselected; SSC is operating without slave select input functionality. 001 _B <u>SLSI1</u> input line is selected for operation 010 _B <u>SLSI2</u> input line is selected for operation 011 _B <u>SLSI3</u> input line is selected for operation 100 _B <u>SLSI4</u> input line is selected for operation 101 _B <u>SLSI5</u> input line is selected for operation 110 _B <u>SLSI6</u> input line is selected for operation 111 _B <u>SLSI7</u> input line is selected for operation
STIP	8	rw	Slave Transmit Idle State Polarity This bit defines the logic level of the slave mode transmit signal MRST when the SSC is deselected (PISEL.SLSIS = 0). 0 MRST = 0 when SSC is deselected in slave mode. 1 MRST = 1 when SSC is deselected in slave mode.
0	[7:6], [31:9]	r	Reserved ; returns 0 if read; should be written with 0.

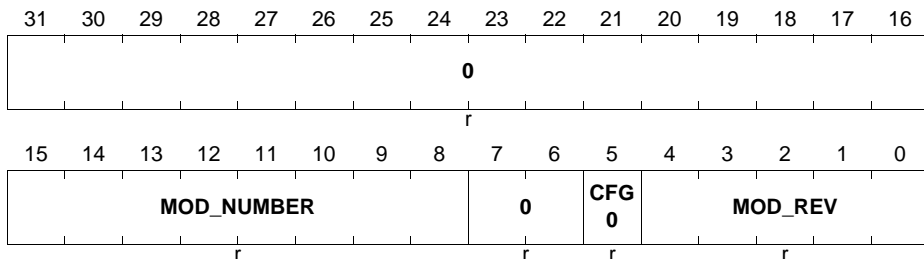
Synchronous Serial Interface (SSC)

ID

Module Identification Register

(08_H)

Reset Value: 0000 4510_H



Field	Bits	Type	Description
MOD_REV	[4:0]	r	Module Revision Number MOD_REV defines the module revision number. The value of a module revision starts with 1 _H (first revision). Current Module Revision is 10 _H
CFG0	5	r	Configuration This bit field indicates whether FIFOs are available or not. 0 No FIFOs available 1 FIFOs available
MOD_NUMBER	[15:8]	r	Module Number Value This bit field defines the SSC module identification number (= 45 _H).
0	[7:6], [31:16]	r	Reserved ; returns 0 if read; should be written with 0.

Synchronous Serial Interface (SSC)

The operating modes of the SSC is controlled by the control register CON. This register contains control bits for mode and error check selection.

CON

Control Register

(10_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	MS	0	A REN	BEN	PEN	REN	TEN	LB	PO	PH	HB	BM			
rw	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

Field	Bits	Type	Description
BM	[3:0]	rw	Data Width Selection BM defines the number of data bits of the serial frame. 0000 _B Reserved; do not use this combination. 0001 _B to 1111 _B Transfer Data Width is 2 ... 16 bit (<BM> + 1)
HB	4	rw	Heading Control 0 Transmit/Receive LSB First 1 Transmit/Receive MSB First
PH	5	rw	Clock Phase Control 0 Shift transmit data on the leading clock edge, latch on trailing edge 1 Latch receive data on leading clock edge, shift on trailing clock edge (Master Mode only ¹⁾)
PO	6	rw	Clock Polarity Control 0 Idle clock line is low, the leading clock edge is low-to-high transition 1 Idle clock line is high, the leading clock edge is high-to-low transition
LB	7	rw	Loop Back Control 0 Normal output 1 Receive input is connected with transmit output (Half-duplex Mode)

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
TEN	8	rw	Transmit Error Enable 0 Ignore transmit errors 1 Check transmit errors
REN	9	rw	Receive Error Enable 0 Ignore receive errors 1 Check receive errors
PEN	10	rw	Phase Error Enable 0 Ignore phase errors 1 Check phase errors
BEN	11	rw	Baud Rate Error Enable 0 Ignore baud rate errors 1 Check baud rate errors
AREN	12	rw	Automatic Reset Enable 0 No additional action upon a baud rate error 1 SSC is automatically reset on a baud rate error
MS	14	rw	Master Select 0 Slave Mode. Operate on shift clock received via SCLK 1 Master Mode. Generate shift clock and output it via SCLK The inverted state of this bit is available at the M/S select output line.
EN	15	rw	Enable Bit 0 Transmission and reception is disabled. 1 Transmission and reception is enabled. This bit is available at the enable output line.
0	13, [31:16]	r	Reserved ; returns 0 if read; should be written with 0.

¹⁾ CON.PH=1 may not be configured in slave mode, because the first bit to be transmitted (LSB or MSB of the transmit buffer TB) on the MRST line after the falling edge of the SLSIx is replaced by the **PISEL.STIP** value. In master mode in most cases the first received bit is invalid.

Synchronous Serial Interface (SSC)

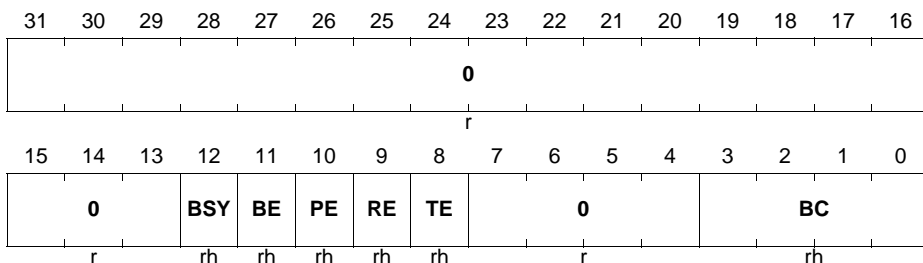
The status register STAT contains status flags for error identification, the busy flag, and a bit field that indicates the current shift counter status.

STAT

Status Register

(28_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
BC	[3:0]	rh	Bit Count Status BC indicates the current status of the shift counter. The shift counter is updated with every shifted bit.
TE	8	rh	Transmit Error Flag 0 No error 1 Transfer starts with the slave's transmit buffer not being updated
RE	9	rh	Receive Error Flag 0 No error 1 Reception completed before the receive buffer was read
PE	10	rh	Phase Error Flag 0 No error 1 Received data changes around the sampling clock edge
BE	11	rh	Baud Rate Error Flag 0 No error 1 More than factor 2 or 0.5 between slave's actual and expected baud rate
BSY	12	rh	Busy Flag BSY is set while a transfer is in progress.

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
0	[7:4], [31:13]	r	Reserved ; returns 0 if read; should be written with 0.

The error flag modification register EFM is required for resetting or setting the four error flags which are located in register STAT.

Synchronous Serial Interface (SSC)

EFM

Error Flag Modification Register

(2C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SET BE	SET PE	SET RE	SET TE	CLR BE	CLR PE	CLR RE	CLR TE	0							
w	w	w	w	w	w	w	w	r							

Field	Bits	Type	Description
CLRTE	8	w	Clear Transmit Error Flag Bit 0 No effect. 1 Bit STAT.TE is cleared. Bit is always read as 0.
CLRRE	9	w	Clear Receive Error Flag Bit 0 No effect. 1 Bit STAT.RE is cleared. Bit is always read as 0.
CLRPE	10	w	Clear Phase Error Flag Bit 0 No effect. 1 Bit STAT.PE is cleared. Bit is always read as 0.
CLRBE	11	w	Clear Baud Rate Error Flag Bit 0 No effect. 1 Bit STAT.BE is cleared. Bit is always read as 0.
SETTE	12	w	Set Transmit Error Flag Bit 0 No effect. 1 Bit STAT.TE is set. Bit is always read as 0.
SETRE	13	w	Set Receive Error Flag Bit 0 No effect. 1 Bit STAT.RE is set. Bit is always read as 0.

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
SETPE	14	w	Set Phase Error Flag Bit 0 No effect. 1 Bit STAT.PE is set. Bit is always read as 0.
SETBE	15	w	Set Baud Rate Error Flag Bit 0 No effect. 1 Bit STAT.BE is set. Bit is always read as 0.
0	[7:0], [31:16]	r	Reserved ; returns 0 if read; should be written with 0.

Note: When the set and clear bit for an error flag is set at the same time during an EFM write operation (e.g SETPE = CLRPE = 1), the error flag in STAT is not affected.

Synchronous Serial Interface (SSC)

The chip select control register controls the operation of the chip select generation unit.

SSOC

Slave Select Output Control Register (18_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OEN	OEN	OEN	OEN	OEN	OEN	OEN	OEN	AOL	AOL	AOL	AOL	AOL	AOL	AOL	AOL
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
AOLn (n = 0-7)	n	rw	Active Output Level 0 SLSON is at low-level during the chip select active time t_{SLSOACT} . The high level is the inactive level of SLSON. 1 SLSON line n is at high level during the chip select active time t_{SLSOACT} . The low-level is the inactive level of SLSON.
OENn (n = 0-7)	8 + n	rw	Output n Enable Control 0 SLSON output is disabled; SLSON is always at inactive level as defined by AOLn. 1 SLSON output is enabled
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Note: This register is buffered during a transfer.

Synchronous Serial Interface (SSC)

The chip select control register controls the operation of the chip select generation unit.

SSOTC

Slave Select Output Timing Control Register (1C_H) **Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0							SLS 07 MOD	0	INACT		TRAIL		LEAD		
r							rw	r	rw		rw		rw		

Field	Bits	Type	Description
LEAD	[1:0]	rw	Slave Output Select Leading Delay This bit field defines the number of leading 00 _B Zero leading delay clock cycles selected 01 _B One leading delay clock cycle selected 10 _B Two leading delay clock cycles selected 11 _B Three leading delay clock cycles selected A leading delay clock cycle is always a multiple of an SCLK shift clock period.
TRAIL	[3:2]	rw	Slave Output Select Trailing Delay 00 _B Zero trailing delay clock cycles selected 01 _B One trailing delay clock cycle selected 10 _B Two trailing delay clock cycles selected 11 _B Three trailing delay clock cycles selected A trailing delay clock cycle is always a multiple of an SCLK shift clock period.
INACT	[5:4]	rw	Slave Output Select Inactive Delay 00 _B Zero inactive delay clock cycles selected 01 _B One inactive delay clock cycle selected 10 _B Two inactive delay clock cycles selected 11 _B Three inactive delay clock cycles selected A inactive delay clock cycle is always a multiple of an SCLK shift clock period.

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
SLSO7MOD	8	rw	SLSO7 Delayed Mode Selection This bit selects the delayed mode for the SLSO7 slave select output. 0 Normal mode selected for SLSO7 1 Delayed mode selected for SLSO7
0	[7:6], [31:9]	r	Reserved ; read as 0; should be written with 0.

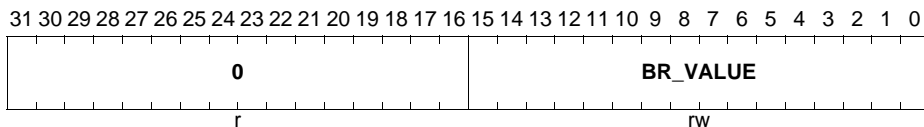
Note: This register is buffered during a transfer.

Synchronous Serial Interface (SSC)

The SSC baud rate timer reload register BR contains the 16-bit reload value for the baud rate timer.

BR

Baud Rate Timer Reload Register **(14_H)** **Reset Value: 0000 0000_H**



Field	Bits	Type	Description
BR_VALUE	[15:0]	rw	Baud Rate Timer/Reload Register Value Reading BR returns the 16-bit content of the baud rate timer. Writing BR loads the baud rate timer reload register with BR_VALUE.
0	[31:16]	r	Reserved ; returns 0 if read; should be written with 0.

Synchronous Serial Interface (SSC)

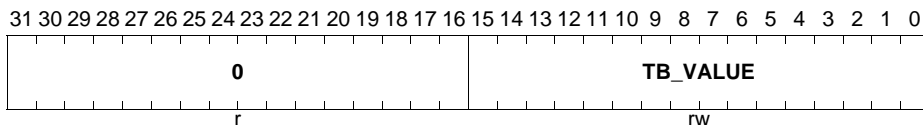
The SSC transmit buffer register TB contains the transmit data value.

TB

Transmit Buffer Register

(20_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
TB_VALUE	[15:0]	rw	Transmit Data Register Value TB_VALUE is the data value to be transmitted. Unselected bits of TB are ignored during transmission.
0	[31:16]	r	Reserved ; returns 0 if read; should be written with 0.

Synchronous Serial Interface (SSC)

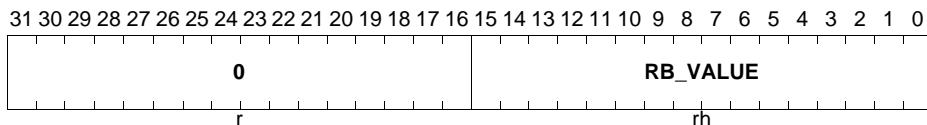
The SSC receive buffer register RB contains the receive data value.

RB

Receive Buffer Register

(24_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
RB_VALUE	[15:0]	rh	Receive Data Register Value RB contains the received data value RB_VALUE. Unselected bits of RB will be not valid and should be ignored.
0	[31:16]	r	Reserved ; returns 0 if read; should be written with 0.

6.3 SSC Module Implementation

6.3.1 Interfaces of the SSC Module

6.3.1.1 Port Connections of SSC

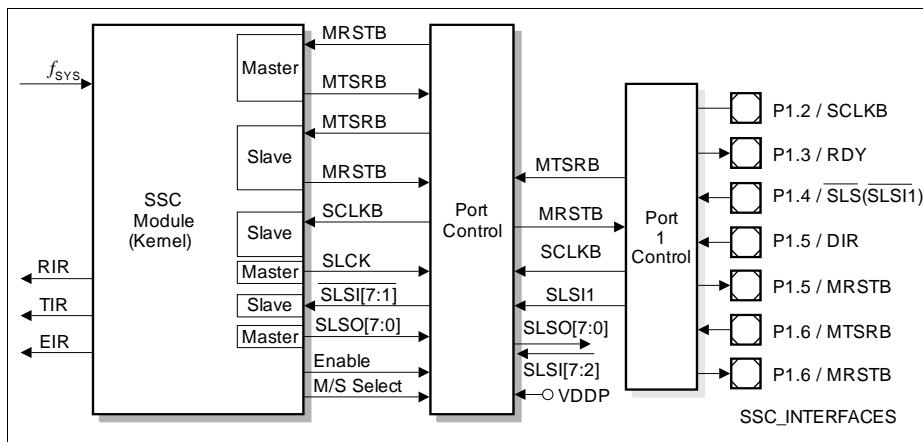


Figure 6-13 SSC Module Implementation and Interconnects

6.3.1.2 Interface signals of SSC

The SSC module has 2 sets of interface signals (MTSRA, MRSTA, SCLKA, MTSRB, MRSTB, SCLKB). During reset the SSC signals are switched to setting A, by programming SSC.PISEL setting B is then initialized by the IME. This means, that

- MTSRB, MRSTB and SCLKB are connected to Port0
- MTSRA and SCLKA remain open
- MRSTA is connected to Vss (Ground)

The SLSOx (slave select output) signals are not used, as the SAK-CIC310-OSMX2HT works as slave. These signals remain open. Concerning the slave select input signals only SLSI1 is used as SLS, SLSI[7:2] are deactivated (connected to V_{DDP}). The IME initializes the SLSI1 input by writing to PISEL.SLSIS.

6.3.1.3 RDY signal

An additional output signal has been added to the standard SSC interface [Figure 6-1](#). This signal "RDY" indicates when the SSC slave device is ready to be accessed. It can be used

Synchronous Serial Interface (SSC)

- To detect an overload situation of the SAK-CIC310-OSMX2HT because of too many host accesses to the module.
- To delay the host, if the SAK-CIC310-OSMX2HT system clock is too slow in comparison to the SSC clock
- To indicate an error situation to the host
- As soon as the falling edge of the $\overline{\text{SLS}}$ input signal has been detected, output RDY becomes active ('1'), indicating that the Standalone SAK-CIC310-OSMX2HT device is ready for data exchange. The detection of the first clock edge for the current halfword transfer resets output RDY.
- After the complete transaction of the current halfword output RDY is set again to indicate anew the ready-state of the slave. Then the next transaction can take place.
- Furthermore it is necessary, that the slave has a possibility to delay the host in case, that it should send a value not yet available or in case, that it should process a received value while still busy.
- As soon as input $\text{SLS} = 1$ has been detected, output RDY is reset. If this is detected before having set RDY, RDY will not be set. The communication sequence is finished.

6.3.1.4 DIR Signal

An additional input signal has been added to the standard SSC interface [Figure 6-1](#). This signal "DIR" indicates in half-duplex mode when the SSC slave device should drive the common communication line.

- As soon as the falling edge of the DIR input signal has been detected, the alternative Port Control field (PC6B) becomes active, indicating that the Standalone SAK-CIC310-OSMX2HT device should drive the common MRSTB/MTSRB pin.

6.3.1.5 Connecting 2 or more slaves to 1 host

If more than only one slave is connected to a SSC master, only one of them may be selected at a time. The master enables one of the connected slaves with the associated slave select output signals SLSO_x . The other output signals of the master (MSTR and SCLK) are broadcast and connected to each slave. The output signals of the slaves (MRST and RDY), which are inputs of the master, have to be activated by the selected slave and have to be deactivated by all other slaves. In this way it is possible to directly connect the incoming signals at the master's input terminal. The deactivation of the outputs (MRST and RDY) of the deselected slaves is accomplished by deactivating the output stage on the associated pads. This deactivation takes place, if $\text{MODE} = 11$ and $\overline{\text{SLS}} = 1$, i.e. SSC selected and slave not selected ([Chapter 1.3](#)).

$\text{IOCRB} = \overline{\text{SLS}}$ and $\text{MODE}=11$

Two further bit fields PC3B in register P1_IOCR0 for port P1.3 and PC5B in register P1_IOCR4 for port P1.5 are defined with the same definition PCx as in [Table 9-2](#). If the upper condition is true, the signals OCRB3 and OCRB5 become true, then the new bit

Synchronous Serial Interface (SSC)

fields PC3B respectively PC5B define the GPIO port behavior, otherwise the regular PC3 respectively PC5 defines the GPIO port behavior. These new bit fields are programmable and should be programmed to an input-function in this case. The port thus becomes input, whenever SLS goes high and the slave is deselected.

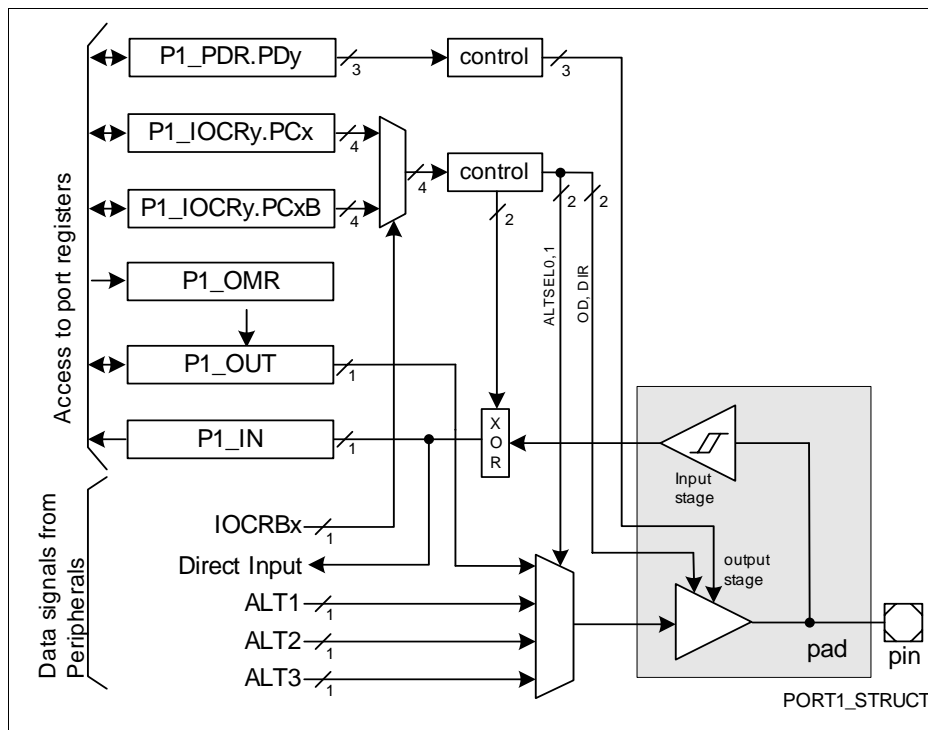


Figure 6-14 Port1 Structure with alternative IO-control

6.3.1.6 Half-Duplex Control

Because the data inputs and outputs are connected together, a transmitting device will clock in its own data at the input pin (MRST for a master device, MTSR for a slave). In this way, any corruption is detected on the common data exchange line where the received data is not equal to the transmitted data.

The master enables one of the connected slaves with the associated direction select input signal DIR. The output signal of the slaves (MRST), which is a input of the master, is activated by the DIR signal and the input signal of the slave (MTSR), which is a output of the master, is deactivated by all other slaves. The user has to take care that the host

Synchronous Serial Interface (SSC)

controller respectively drives the single line. The switching between the output (MRST) and input (MTSR) of the deselected slaves is accomplished by deactivating the output stage on the associated pads. This deactivation takes place, if $MODE = 11$ and $DIR = 1$, i.e. SSC selected and direction selected.

IOCRB = SSC_EERCUM.MSCON and DIR and MODE=11

An additional bit field PC6B in register P1_IOC4 for port P1.6 is defined with the same definition PCx as in [Table 9-2](#). If the upper condition is true, the signals OCRB6 become true, then the new bit fields PC6B defines the GPIO port behavior, otherwise the regular PC6 defines the GPIO port behavior. This new bit field is programmable and should be programmed to an input-function in this case. The port thus becomes input, whenever DIR goes low.

6.3.2 Error Handling

An error occurs among other things by

- The transmission of a lower or a higher number than 16 bits or by a spike on the SCLK line
- Received baud rate exceeding the limits of the double and the half of the selected baud rate (has to be enable by application software). May not be enabled if communicating to SAK-CIC310-OSMX2HT by an Asynchronous/Synchronous Serial Communication Interface (ASC).
- Any error of the original SSC status register STAT ([Chapter 6-3](#)).

If an error condition occurs, the SSC is no more functional and therefor has to be initialized again as fast as possible. The host is not informed about the error. Even the RDY signal does not give an indication about the error state. So as there is no mean to inform the host, the initialization has to take place automatically on the SAK-CIC310-OSMX2HT. Before the next transmission is started, the RDY signal only goes high, if the auto initialization was successfully completed and all other conditions are true. It is desirable to record occurring errors, this is done in the SSC Cumulative Error Register EERCUM.

Note: This register is located in the SCU and only showed here for a better understanding.

Synchronous Serial Interface (SSC)

SSC_ERRCUM

Cumulative Error Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			MS CON	BE	PE	RE	TE	0				0			
r			rw	rwh	rwh	rwh	rwh	r				r			

Field	Bits	Type	Description
TE	8	rwh	Transmit Error Flag 0 _B No error 1 _B Transfer starts with the slave's transmit buffer not being updated Hardware only sets this error flag and software has to reset flag respectively.
RE	9	rwh	Receive Error Flag 0 _B No error 1 _B Reception completed before the receive buffer was read Hardware only sets this error flag and software has to reset flag respectively.
PE	10	rwh	Phase Error Flag 0 _B No error 1 _B Received data changes around the sampling clock edge Hardware only sets this error flag and software has to reset flag respectively.
BE	11	rwh	Baud Rate Error Flag 0 _B No error 1 _B More than factor 2 or 0.5 between slave's actual and expected baud rate Hardware only sets this error flag and software has to reset flag respectively.

Synchronous Serial Interface (SSC)

Field	Bits	Type	Description
MSCON	12	rw	Master Slave Control 0_B No Half-Duplex Support for SSC. P1_IOC4.PC6 controls the characteristics of P1.6 1_B Half Duplex Support for SSC. Input of P1.5 selects if P1_IOC4.PC6 (P1.5=0) or P1_IOC4.PC6B (P1.5=1) controls the characteristics of P1.6 <i>Note: This bit is only active, if $MODE=11_B$ (SSC Host Interface active).</i>
0	[7:0], [31:13]	r	Reserved ; returns 0 if read; should be written with 0.

The errors are recorded continuously and are accumulated. They remain set until the user clears the flags intentionally. So a new error flag has to be binary ORed into the old state of the register. So starting with a cleared register ERRCON the host can make sure after some time, that no error in the meantime occurred. If an error occurred the register should be cleared again.

6.3.3 Reset Initialization

Even so the SSC is highly flexible, not all features can be used or configured within the SAK-CIC310-OSMX2HT, because the SSC is internally pre configured by the Initialization Move Engine (IME) for:

- The device is configured to run in SSC Slave Mode.
- The transfer data width is 16 bit.
- The LSB is transferred and received first.
- The receiving data is latched on trailing clock edge and shifted within the shift register on leading edge.
- The idle clock line is configured to be high.
- The leading clock edge is configured to be a high-to-low transition.
- Transmit error, receive error, and phase error are activated.
- In case of communication to an ASC communication controller, the baud rate error has to be ignored. Therefore proposal is to configure the SSC to ignore this error. Customer may activate error within the application software.
- The P1.2 port pin is configured as SSC slave select input signal (SSC1_SLSI1).
- The Slave Mode transmit idle polarity is a high level, so driving the MRST to high (1) when SSC is deselected in slave mode.
- The $\overline{SLSI1}$ is selected as Slave Mode slave select (\overline{SLS}) input.
- The SCLKB is selected as Slave Mode clock input.

Synchronous Serial Interface (SSC)

- The MTSRB is selected as Slave Mode receive input.
- The MRSTA is selected as Master Mode Receive Input.
- The P1.6 port pin is configured as SSC clock input signal (SSC_SCLKB).
- The P6.4 port pin is configured as SSC master transmit slave receive input signal (SSC_MTSRB).
- The P1.8 port pin is configured as SSC master receive slave transmit output signal (SSC_MRSTB). Therefore the push/pull function is activated and the output driver characteristic set to strong driver, sharp edge.

Changing these parameters may lead to a complete communication blackout and therefore should be avoided.

6.3.4 Address Map

In the FlexRay Communication Controller, the registers of the SSC module are located in the following address Range:

- Module Base Address = 0000 0900_H
Module End Address = 0000 09FF_H
- Absolute Register Address = Module Base Address + Offset Address
(offset addresses see [Table 6-3](#))

7 Micro Link Serial Bus Interface (MLI)

This chapter describes the micro link serial bus interface of the SAK-CIC310-OSMX2HT. It contains the following sections:

- MLI applications (see [Section 7.1](#))
- Functional description of the MLI kernel (see [Section 7.2](#))
- MLI kernel register descriptions of all MLI kernel specific registers (see [Section 7.3](#))
- SAK-CIC310-OSMX2HT implementation specific details and registers of the MLI module (port connections and control, address decoding, (see [Section 7.4](#))

7.1 MLI Applications

- Data and program exchanging without intervention of CPU or PCP between microcontrollers of the AUDO-NG family or other devices.
- The internal architecture of the block allows the communication between microcontrollers in different clock domains.
- Compatibility with the SSC interface.
- The read mode also allows to ask the other microcontroller for the desired data.
- Resources sharing. It is possible to use resources not available in the microcontroller but present in another one.
- Capability of programming up to four different interrupts in the second microcontroller by sending a command.

[Figure 7-1](#) shows an general overview of the MLI location in the microcontroller and its connection within another MLI.

Micro Link Serial Bus Interface (MLI)

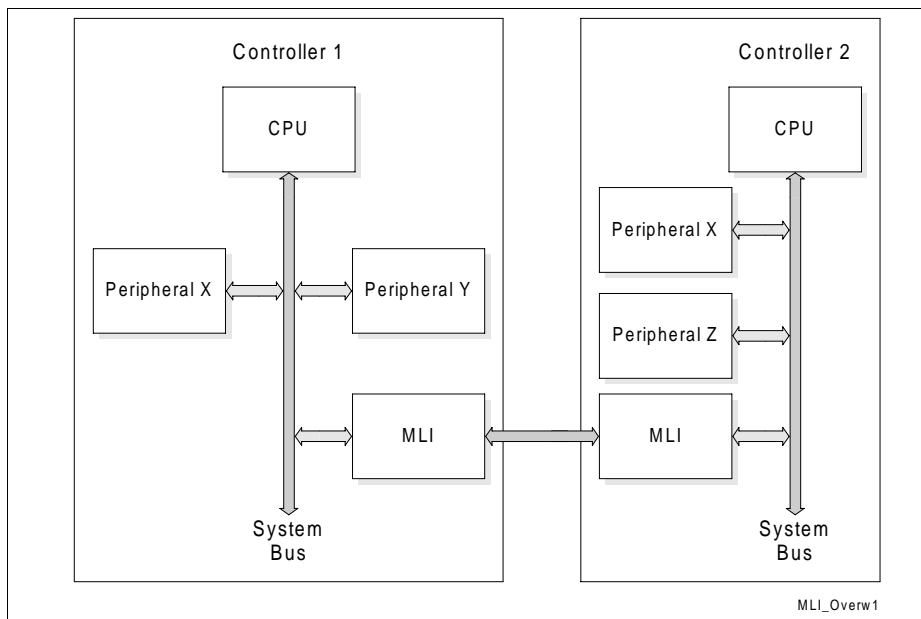


Figure 7-1 Location of MLI in the microcontroller and Connection

Micro Link Serial Bus Interface (MLI)

7.2 MLI Kernel Description

Figure 7-2 shows a global view of all functional blocks of the MLI Module.

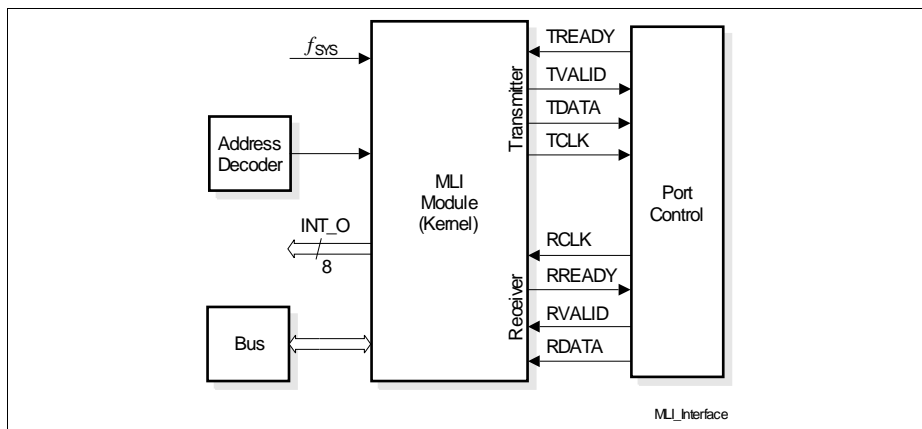


Figure 7-2 General Block Diagram of the MLI Module

7.2.1 Overview

The Micro Link Serial Bus Interface, referenced as MLI in the whole chapter, is dedicated for the serial communication between microcontrollers of the AUDO-NG 32-bit microcontrollers family. The communication is intended to be fast and intelligent due to an address translation system, and it is not necessary to have any special program in the second microcontroller.

An overview of the MLI kernel is shown in **Figure 7-3**.

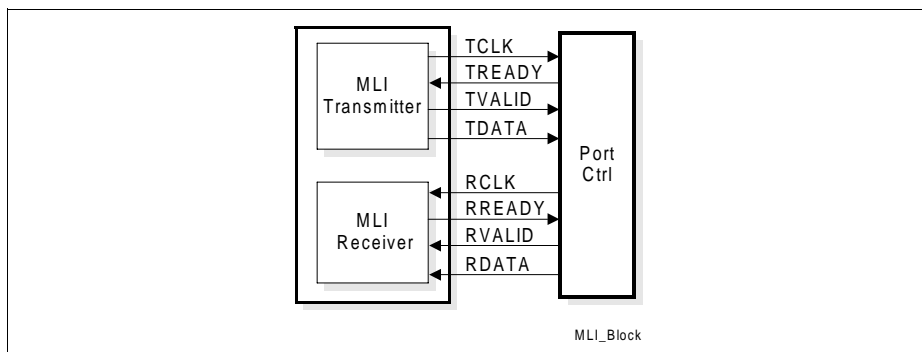


Figure 7-3 MLI Overview

Micro Link Serial Bus Interface (MLI)

Note: The prefixes T and R indicate if the corresponding signals belong to the MLI transmitter or to the MLI receiver.

Features

- Serial communication from the MLI transmitter to MLI receiver of another microcontroller
- Module supports connection of each MLI with up to four MLI from other microcontrollers (see implementation sub-chapter for details for this product)
- Fully transparent read/write access supported (= remote programming)
- Complete address range of target microcontroller available
- Special protocol to transfer data, address offset, or address offset and data
- Error control using a parity bit
- 32-bit, 16-bit, and 8-bit data transfers
- Address offset width: from 1 to 16-bit
- Baud rate max.: $f_{\text{SYS}} / 2$ (symmetric shift clock approach),
baud rate definition by the corresponding fractional divider

7.2.1.1 Naming Conventions**Local and Remote microcontroller**

The names “local” and “remote” microcontroller (device) are assigned to the two partners (microcontrollers with MLI modules) of a serial MLI connection. The microcontroller with an MLI module that operates as a master of the serial MLI connection is defined as local microcontroller. A local microcontroller handles data operations with transfer windows and further initiates all control tasks (control, address, and data transmissions) that are required for the data transfer/request between local and remote microcontroller.

The microcontroller with an MLI module that operates as a slave of the serial MLI connection is defined as remote microcontroller. A remote microcontroller handles data operations with remote windows and executes the tasks that have been assigned/requested by the local microcontroller.

Due to the full duplex operation capability of an MLI module, two serial MLI connections can be installed simultaneously (both transmitters can send a frame to their receivers). This means, each microcontroller with an MLI module is able to operate as local microcontroller as well as remote microcontroller at the same time.

Transfer Window

A transfer window is an address space in the address map of the local microcontroller that is typically not assigned to memories or peripheral units. Transfer windows are always assigned to fixed address space (base address and size) in a specific microcontroller. Each MLI module supports up to four transfer windows with two different window sizes each: four small transfer windows with 8 Kbyte and four large transfer

Micro Link Serial Bus Interface (MLI)

windows with 64 Kbyte. Address and data information that has been written or read to/from transfer windows can be detected and handled by the MLI module of the local microcontroller.

Remote Window

A remote window is an area in the address space of the remote microcontroller. Remote window parameters (base address and window size) are defined and controlled by the local microcontroller. The size of a remote window is defined in a 4-bit coded buffer size parameter that defines the number of variable address bits of a remote window. Each MLI module supports up to four remote windows.

Pipe

A pipe defines the logical connection between an MLI module in the local microcontroller and an MLI module in the remote microcontroller. The logical connection of a pipe maps the transfer window of the transfer window in the local microcontroller to its corresponding remote window in the remote microcontroller. The MLI module supports four pipes.

7.2.1.2 MLI Communication Principles

The communication principle of the MLI modules allows that data is transferred between a local and a remote microcontroller without intervention of a CPU. Data transfers are always triggered in the local microcontroller by read or write operations to memory locations that are located in a transfer window of the local microcontroller. All control tasks (control, address, and data transmissions) that are required for the data read/write accesses between local and remote microcontroller are handled autonomously by the two MLI modules.

A CPU, DMA, or PCP writing to a location within a MLI transfer window of the local microcontroller is detected by the MLI transmitter. This detection initiates a transfer of the data that has been written to the transfer window from the local microcontroller to the MLI receiver of the remote microcontroller which places the data at an address location in a remote window of the remote microcontroller.

A CPU, DMA, or PCP reading from a location within a MLI transfer window in the local microcontroller returns dummy data and initiates the MLI connection to request data from the remote microcontroller. Data is read in the remote microcontroller by the MLI module from an address location within the remote window and transferred to the local microcontroller where it is stored in its MLI receiver registers. Afterwards, the CPU in the local microcontroller is informed by an interrupt that the requested data is now available and can be read from a register.

Micro Link Serial Bus Interface (MLI)

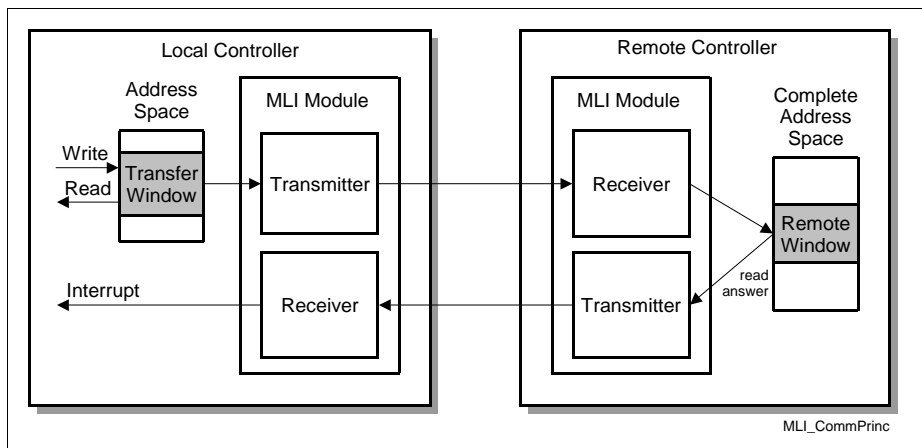


Figure 7-4 MLI Communication Principles

7.2.2 General Description

The communication between both microcontrollers is based in an address translation table that allows the MLI transmitter from the first microcontroller just sending an offset relative to these addresses instead of the full 32-bit address. The consistency for these addresses is guaranteed because the first microcontroller sends all of them to the second microcontroller and this one stores them as explained in [Section 7.2.6.5](#) and in [Section 7.2.6.6](#). Each of the addresses defines a pipe and together with the Buffer Size parameter will define a buffer in the second microcontroller address map. For each MLI transmitter there will be up to four pipes. A pipe may be seen as a logical connection between two microcontrollers.

[Figure 7-7](#) shows the organization in memory of the MLI transfer windows and their possible correspondence in the second microcontroller address map.

Micro Link Serial Bus Interface (MLI)

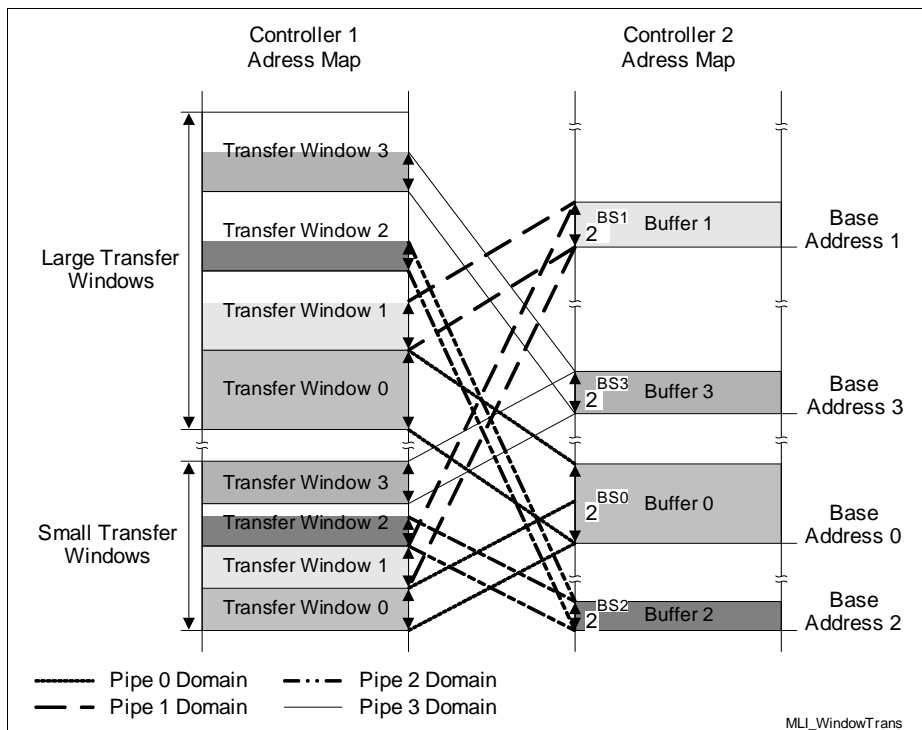


Figure 7-5 Transfer Window Base Address Copy

Note: BS_x is the buffer size of each of the different pipes (where x = 0, 1, 2, 3).

Figure 7-6 illustrates a general overview of a two MLI connected, in which the transfer windows, TW, and the move engine of the receiver have been detailed.

Micro Link Serial Bus Interface (MLI)

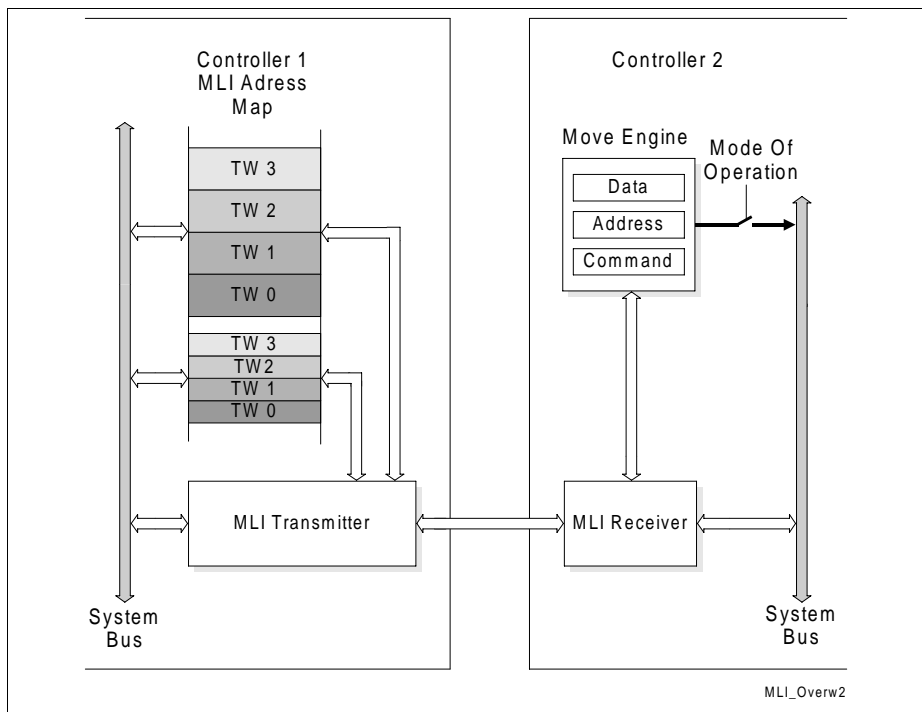


Figure 7-6 MLI Connection Overview

A **CPU, DMA, or PCP writing within a MLI transfer window** of microcontroller 1 leads to a transfer from the MLI transmitter to the MLI receiver of microcontroller 2. The received information (including data and address or the command) is stored in the MLI receiver. There it is available for the CPU of microcontroller 2 or the move can be executed autonomously by the MLI move engine (according to the selected access protection).

A **CPU, DMA, or PCP reading within a transfer window** delivers a dummy value to the respective CPU/DMA. The read request is transferred to the MLI receiver on microcontroller 2. If enabled, the MLI move engine executes the read operation autonomously and the requested data will be sent back to the MLI on microcontroller 1 (by the MLI transmitter on microcontroller 2 to the MLI receiver of microcontroller 1). When this information is available in the MLI module of microcontroller 1, an interrupt can be generated and the CPU (PCP, or a DMA, etc.) of microcontroller 1 can read the requested data.

Micro Link Serial Bus Interface (MLI)

Figure 7-7 shows the process of transmission of the base address for each transfer window.

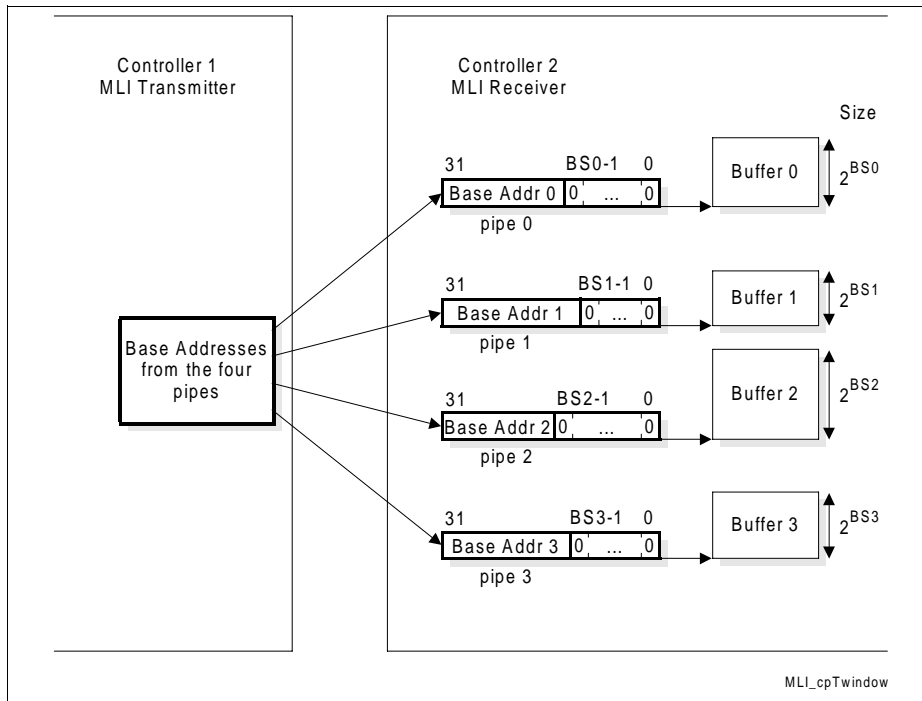


Figure 7-7 Transfer Window Base Address Copy

Note: BS_x is the buffer size of each of the different pipes (where $x = 0, 1, 2, 3$). The selected buffer size must not exceed the size of the targeted transfer window (8 Kbyte for small transfer windows).

Within the offset (its width in bits is the same as indicated in the buffer size), the MLI transmitter from the first microcontroller sends a reference to the pipe in use. When the MLI receiver obtained this data it will know what is the absolute address by simply concatenating the offset to the base address of the pipe.

Note: A pipe should always be accessed by either its small transfer window or the corresponding large transfer window. Mixing accesses via both window types to the same pipe is possible but not recommended (optimized frames require a single transfer window type). The used transfer window type of a pipe can be different from those of the other pipes.

Micro Link Serial Bus Interface (MLI)

Figure 7-8 illustrates the address translation process from the first microcontroller to the second one.

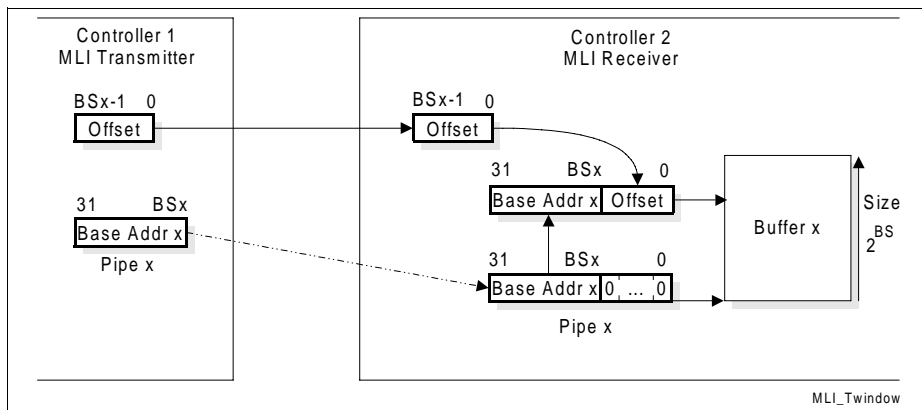


Figure 7-8 Address Translation Process

The kernel MLI includes an optimized mode to transfer data blocks. Whenever the MLI transmitter detects that the new address and the previous one follow a predictable scheme, it will send just the data reducing this way the bits to be transferred. This mode is based in a prediction method of the new address in the MLI receiver of the second microcontroller. In fact the MLI receiver will automatically update the address, as explained in the next sections.

From the point of view of the connection, each MLI transmitter will have four READY, four VALID, one CLK and one DATA possible connections with the external world. This will provide the possibility of connecting each MLI transmitter with up to four MLI receivers in other microcontrollers although the MLI transmitter will not have the possibility of transferring data to two or more different MLI receivers at the same time.

Each MLI receiver will have one READY, four VALID, four CLK and four DATA possible connections with the external world. With this scheme, each MLI receiver could be connected with up to four MLI transmitters in other microcontrollers although the MLI receiver will not have the be possibility of receiving data from two or more different MLI transmitters at the same time.

Figure 7-9 illustrates an example of connection between two MLI.

Micro Link Serial Bus Interface (MLI)

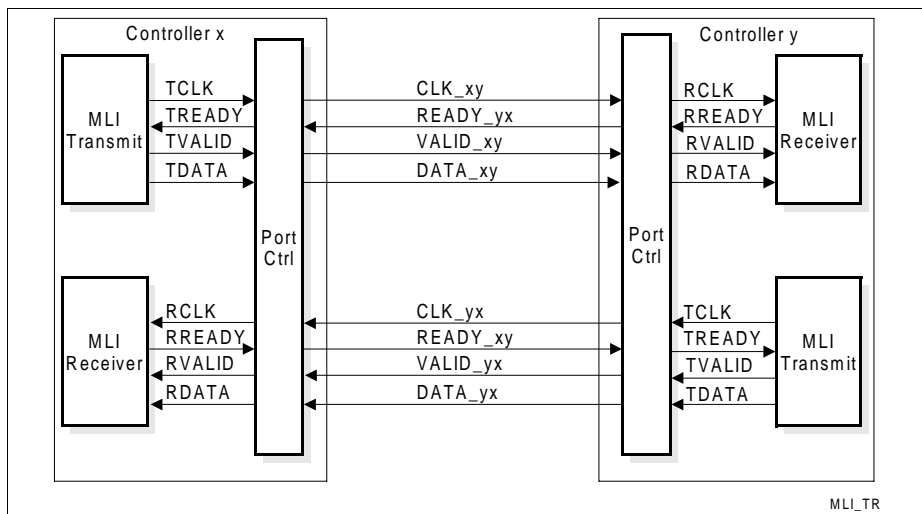


Figure 7-9 MLI Transmitter - Receiver Connection

Note: The suffixes x and y indicate the source and destination of the signals.

Figure 7-10 illustrates an example of connection between three different MLI.

Micro Link Serial Bus Interface (MLI)

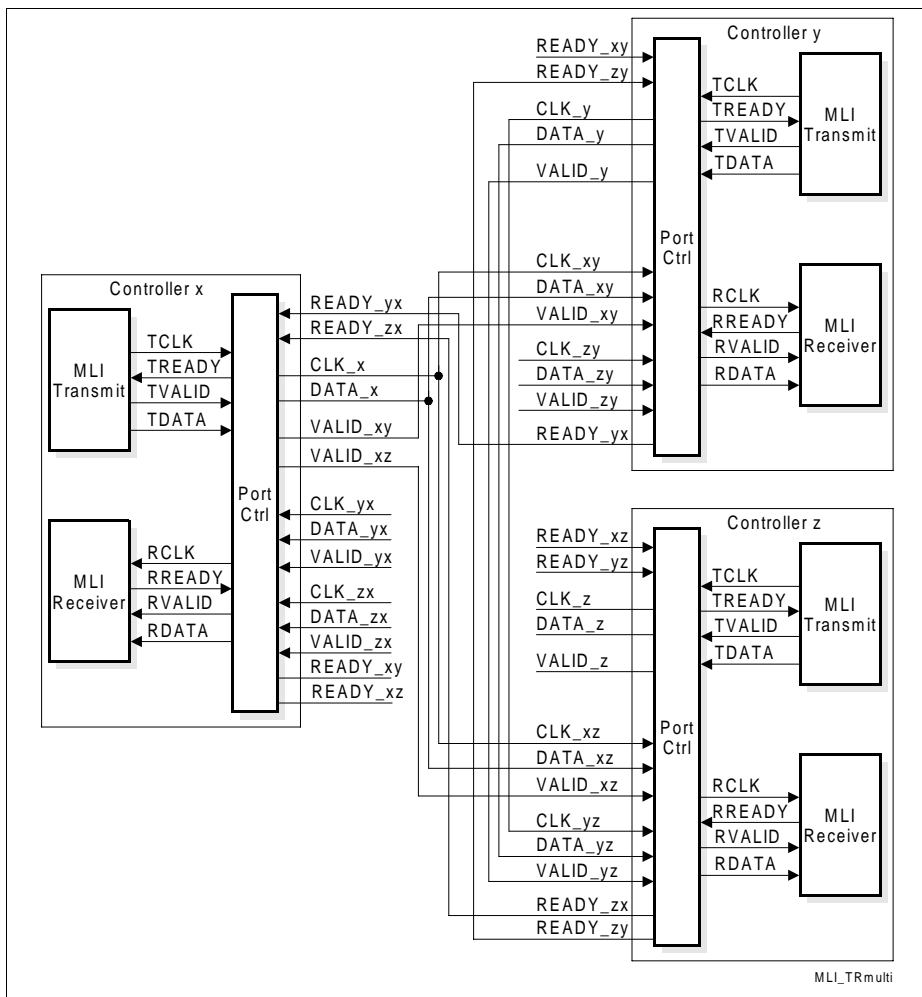


Figure 7-10 MLI Transmitter - Receiver Connection

*Note: The suffixes x, y and z indicate the source and destination of the signals. For instance CLK_xz connects the clock signal CLK_x in the MLI transmitter x and finishes in the MLI receiver z. The signals selection is made by programming the **OICR** register.*

7.2.3 Handshake Description

The signal naming is referring to [Chapter 7-9](#). The transmission may start whenever the MLI receiver in the remote microcontroller is ready to receive data, i.e. receiver output signal RREADY is set high, so the transmitter input signal TREADY becomes high. When the MLI transmitter of the local microcontroller starts a transmission, it sets the transmitter output signal TVALID high and it holds it as long as it sends the data. When the rising edge of the receiver input signal RVALID is detected by the MLI receiver in the remote microcontroller, the receiver output signal RREADY signal is cleared to low level again.

When the transmission is finished the MLI transmitter clears the TVALID signal and checks if the TREADY input signal is low again. This indicates to the local MLI transmitter that the remote MLI receiver has acknowledged the transmission. The local transmitter clears the ready delay counter ([TSTATR.RDC](#)) and it starts counting TCLK clock periods. The counter is stopped when the local MLI transmitter detects that the TREADY signal is at high level again or in case the counter overflows.

If the remote MLI receiver sets high the RREADY signal again after a number of TCLK/RCLK clock periods less than a programmed number (maximum delay for parity, [TCR.MDP](#) in the local microcontroller), this will indicate to the local MLI transmitter that the remote MLI receiver is prepared for a new transmission and that the previous one was received there without parity error.

A detailed explanation about the handshake timing and clock domains in the local MLI transmitter and the remote receiver may be found in [Section 7.2.10](#).

The handshaking in a situation without error is shown in [Figure 7-11](#).

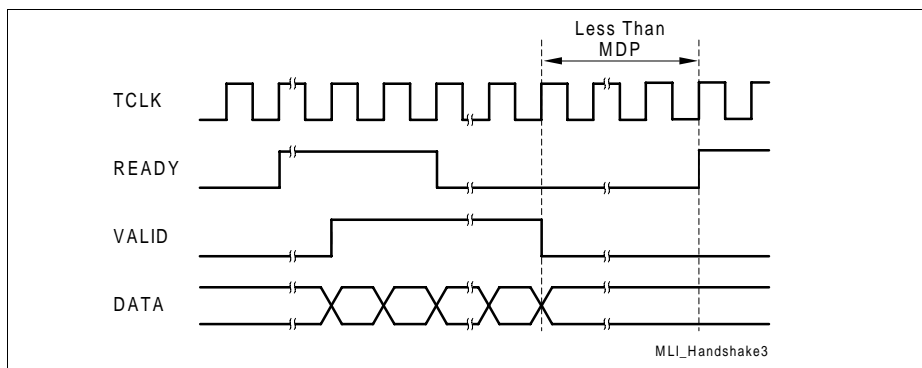


Figure 7-11 MLI Transmitter - Receiver Handshake in Transfer Without Error

Note: The signals are seen from the MLI transmitter in the local microcontroller. When VALID is not asserted the TDATA line will have only a value (one or zero) depending on the programmed value in [TCR.DNT](#) and its chosen polarity.

Micro Link Serial Bus Interface (MLI)

The delay between the falling edge of VALID and the rising edge of READY is measured by the ready delay counter **TSTATR.RDC**. This value is compared to **TCR.MDP** on the transmitter side in order to detect when the receiver has signaled a parity error.

Figure 7-12 illustrates a situation of non acknowledge. The Ready signal remains high when the VALID is set low-level again.

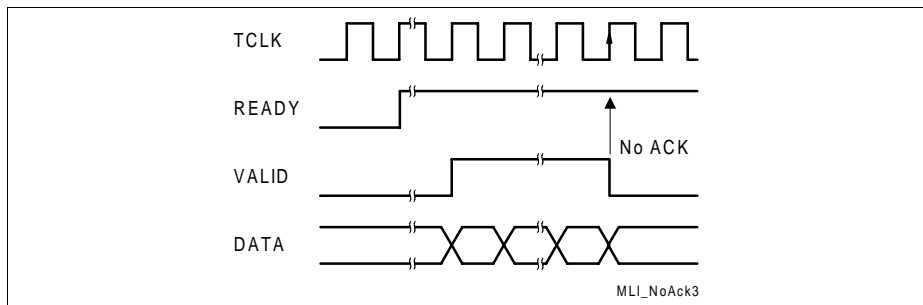


Figure 7-12 Non Acknowledge Situation

Note: The signals are seen from the MLI transmitter in the first microcontroller.

If the READY signal raises after a number of TCLK clock periods (measured by **TSTATR.RDC**) is greater than the maximum delay for parity programmed value (**TCR.MDP**), then this situation will be interpreted as parity error, as shown in **Figure 7-13**.

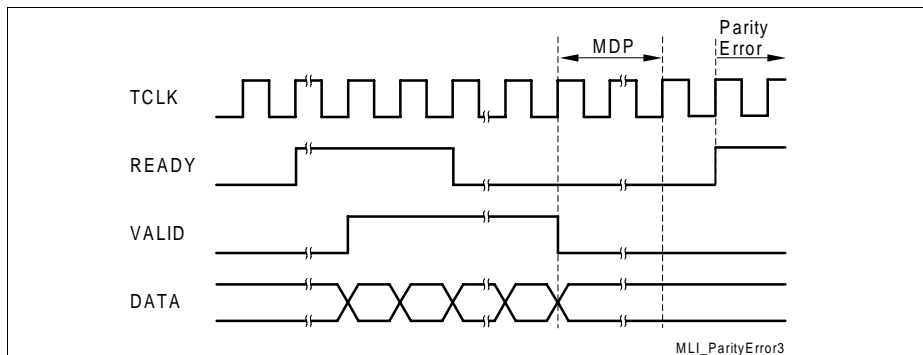


Figure 7-13 Parity Error Situation

Note: The signals are seen from the MLI transmitter in the first microcontroller.

Micro Link Serial Bus Interface (MLI)

If a non acknowledge persists, the MLI transmitter will keep on counting the number of non acknowledge errors and if a maximum number is reached then a time out error is produced. **Figure 7-14** illustrates this situation.

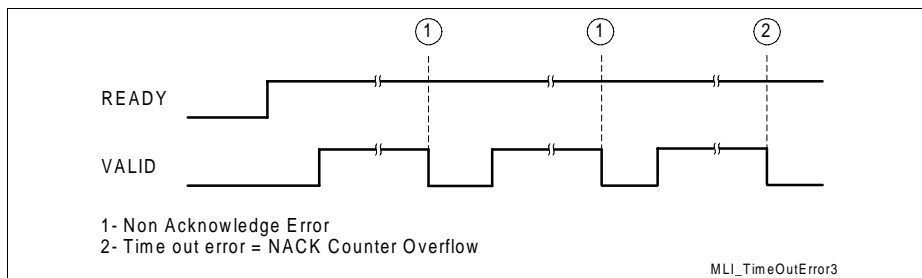


Figure 7-14 Time Out Error Situation

Note: The signals are seen from the MLI transmitter in the first microcontroller.

All the error situations and the actuations taken by both parts are explained in **Section 7.2.6.8** and **Section 7.2.7.5**.

7.2.4 Startup Procedure

During the startup procedure of the MLI, an appropriate value for the maximum delay for parity signaling must be set up in the receiver and in the transmitter. Therefore, the following actions must be taken:

- The overall loop delay (total propagation delay between the transmitter and the receiver, including the output and the input driver delay, the line propagation and the synchronization time) must be measured.
- The appropriate MDP value must be programmed in the transmitter and then transferred to the receiver (automatic receiver setup by a command frame on pipe 1).
- Dummy frames with parity error and without parity error must be sent in order to check for a correct reply of the receiver.
- If the parity error signaling is working correctly, the setup is finished and normal frame traffic can be started.

For the measurement of the overall loop delay a dummy frame is sent to the receiver and the time is measured between the falling edge of the VALID signal and the rising edge of the READY signal (if the READY signal does not rise after a certain while, the receiver might be defect, not correctly connected or not powered).

The signaling of the READY signal on the receiver side takes place in the clock domain of the transmit clock. The reset value of the **TCR.MDP** is 0 in the transmitter and in the receiver and as a result, the READY signal will be set high immediately after receiving the parity bit by the receiver. In this case (MDP = 0), the signaling of READY is identical

Micro Link Serial Bus Interface (MLI)

for a parity error and for a correct frame. Therefore, the number of transmit clock cycles between VALID becoming 0 until READY becoming 1 represents the overall loop delay. It is indicated by the counter bit field **TSTATR.RDC** (ready delay count). This bit field is a counter starting from 0 each time VALID becomes low (1 to 0 transition) and that stops when READY becomes 1 (0 to 1 transition). It holds the value until the next 1 to 0 transition of VALID is reached. This value can be read out to determine an appropriate MDP value.

The desired value for MDP has to be transferred to the receiver. Therefore, the command pipe 1 can be used. A frame of command pipe 1 transfers the value and the receiver stores it automatically as its delay for parity error (**RCR.DPE**). In order to verify the correct setting of DPE of the receiver, dummy frames are transferred with and without parity error. The parity generation of the receiver always starts with 0 and this value is toggled each time a 1 is received. This result is compared to the received parity bit (the received parity bit does not modify the receivers parity check bit). The parity start bit of the transmitter can be programmed. Assuming a correct transfer, the start value of 1 in the transmitter, will lead to a parity error detection on receiver side. This event is signaled by the receiver by setting READY to 1 after the time indicated by **RCR.DPE** (in the receiver) has elapsed. By reading the value of **TSTATR.RDC**, the transmitter SW can detect if the receiver's DPE has been set up correctly. In case of an error, the transfer of DPE by the command pipe 1 has to be started again, until the results are correct.

All these setup actions should take place while the MLI move engine of the receiver is switched off (automatic mode disabled, receiver in listen mode). The complete setup can be done under the control of the transmitter. A special SW on receiver side is not required. If the required values are known on both sides (transmitting and receiving microcontroller), the normal transfers can start.

Note: A dummy frame can be any frame that does not lead to a HW action in the receiver. In listen mode, the CPU on receiver side should ignore the reception of dummy frames. In order to start normal operation, the transmitter can switch on the automatic mode of the receiver's move engine or send a command via pipe3 that is then taken into account by the receiver's CPU.

7.2.5 MLI Kernel and MLI Interface Logical Connection

The MLI transmitter will have the possibility of getting 32 address bits, 32 bits of data, a selection for read or write operation, and a selection of the transmission window. The MLI receiver may provide the MLI interface 32 bits of address, 32 bits of data and the type of the transaction (read or write). In addition there will be a signal to request the DMA from the MLI receiver.

Figure 7-15 shows a graphical representation of this interface.

Micro Link Serial Bus Interface (MLI)

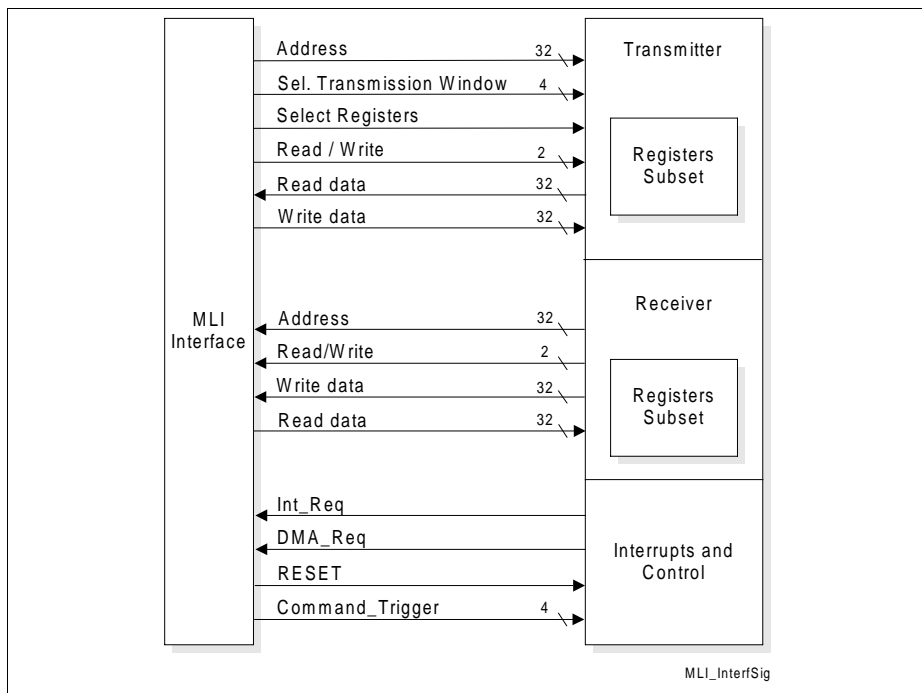


Figure 7-15 Signals Between MLI Interface and MLI Kernel

7.2.6 MLI Transmitter

7.2.6.1 MLI Transmitter Reset

After the hardware reset the MLI transmitter will be in transmitter off mode as it will be explained in [Section 7.2.6.2](#). In this state the transmitter will not raise its VALID signal and therefore no transaction will be performed.

7.2.6.2 MLI Transmitter Operation Modes

By programming the MLI transmitter control register it is possible to set its operation mode. In [Table 7-1](#) are shown all the possible modes of the MLI transmitter, depending

Micro Link Serial Bus Interface (MLI)

on the values of the mode of operation parameter (**TCR.MOD**). The characteristics of the different modes are explained below.

Table 7-1 MLI Transmitter Operation Modes

TCR.MOD	Mode of Operation
0	MLI transmitter off
1	MLI transmitter on

Transmitter Off

This is the mode in which the MLI transmitter is after the hardware reset. It has the following characteristics:

- The VALID signal is not activated. As a consequence, no transfer is supported.

Transmitter On

Its characteristics are the following ones:

- The whole functionality of the MLI transmitter is available.

Note: The next paragraphs will explain the whole MLI transmitter functionality and its interfaces arbitration for the transmitter on operation mode. In every case must be taken into account the operation mode in which the MLI transmitter is programmed and consider the limitations introduced by each of the explained modes.

7.2.6.3 Internal Architecture And Interface Signals

Figure 7-16 illustrates the internal architecture of the MLI transmitter:

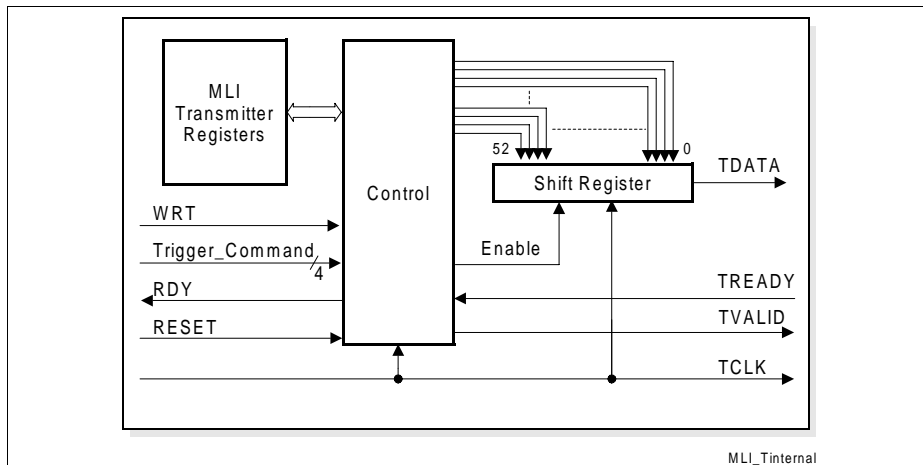


Figure 7-16 MLI Transmitter Internal Architecture

Note: The letter “x” indicates the pipe number, from 0 to 3.

The signal RDY is used to notify the microcontroller that the MLI transmitter is prepared to receive new information in the data and address offset registers (TPxDATAR and TPxAOFR). The WRT signal indicates while active, that a write access is performed in the MLI registers. The enable signal will control the shift register.

The four trigger command lines, will program via hardware the MLI transmitter to send up to four different commands.

The signals TDATA, TREADY, TVALID and TCLK will follow the scheme explained in [Section 7.2.3](#).

The MLI transmitter will operate as an information moving agent between two microcontrollers. It will receive through its microcontroller interface side, address offset and data, only address offset or only data. This values will be written in TPxDATAR and TPxAOFR registers by software via the DMA switch, or by the DMA itself.

The MLI transmitter will keep track of the next information:

- Current address offset in the pipe (TPxAOFR, where x denotes the pipe number)
- Width of the current address offset in each of the pipe's address offset registers (TPxSTATR.BS, where x denotes the pipe number)
- Current data in the pipe (TPxDATAR, where x denotes the pipe number)

Micro Link Serial Bus Interface (MLI)

- Width of the current data received in each of the pipe's data registers (TPxSTATR.DW, where x denotes the pipe number)

A complete list of the MLI transmitter registers may be found in [Section 7.3.2](#).

7.2.6.4 Transmission Format

The MLI transmitter first transmits four bits as header (denoted as H) that contains information about the mode of transfer and which is the current pipe in use.

The first two bits in the header will determine what type of transmission it is and it will be referenced as frame code (denoted as FC in the whole chapter). These two bits together with the number of bits of the frame will determine the type of transmission. [Table 7-2](#) shows its encoding.

Table 7-2 First Two Bits Header Encoding. Frame Code

Header bits 0 and 1 (FC)	Type of Frame
00 _B	Copy base address frame
01 _B	Write in offset and data frame or Discrete read frame
10 _B	Command frame or Answer frame
11 _B	Optimized write frame or Optimized read frame

The second two bits field, indicate which is the current pipe in use. This two bits will be referenced as pipe number (PN) in the whole chapter. [Table 7-3](#) shows its encoding.

Table 7-3 Second Two Bits Header Encoding. Pipe Number

Header bits 2 and 3, (PN)	Pipe in Use
00 _B	Pipe 0
01 _B	Pipe 1
10 _B	Pipe 2
11 _B	Pipe 3

After this header the MLI transmitter sends the information as it will be explained in [Section 7.2.6.5](#). In every case a parity bit will be transmitted in the last position of the frame (p). [Figure 7-17](#) illustrates the parts of the transmission frame.

Micro Link Serial Bus Interface (MLI)

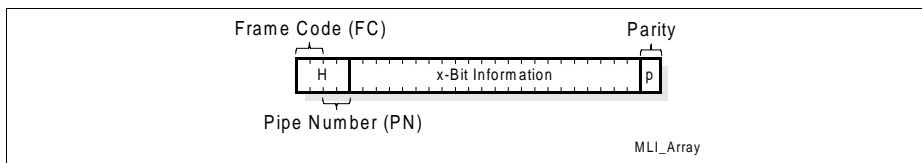


Figure 7-17 Parts of the Transmission Frame

7.2.6.5 Transmission Modes

In this section are explained the different modes in transmission, indicating for each one:

- The frame composition
- Where are the values of each field of the frame taken from
- The number of bits that each frame has

In [Section 7.2.6.6](#) will be explained how the MLI transmitter chooses between these different options of transmission.

Copy Base Address Frame

Its frame code is 00_B. This mode will allow the MLI receiver in the other microcontroller to know the base addresses and the buffer size of each transfer window. The MLI transmitter sends the frame illustrated in [Figure 7-18](#).

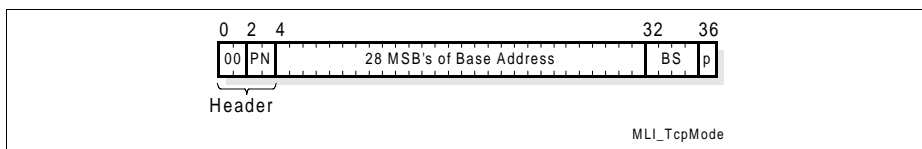


Figure 7-18 Copy Base Address Frame

The frame contains the 28 more significant bits of the base address and the buffer size of this transfer window (the 4 bits denoted as BS). [Table 7-4](#) illustrates where each of the fields of the frame are taken from.

Table 7-4 Storage of the Values Used in the Frame

Field	Value Taken From
PN	TRSTATR .PN
Base Address	28 MSB's of TCBAR
Buffer Size	TPxSTATR.BS

Micro Link Serial Bus Interface (MLI)

Note: x indicates the pipe number, $x = 0, 1, 2, 3$.

The number of bits transmitted in this mode will be as shown in [Table 7-5](#).

Table 7-5 Number of Bits In Copy Base Address Frame

Header	Base Address	Parity	Total
4 bits	32 bits	1 bit	37 bits

Command Frame

Its frame code is 10_B . The MLI transmitter sends the following frame:

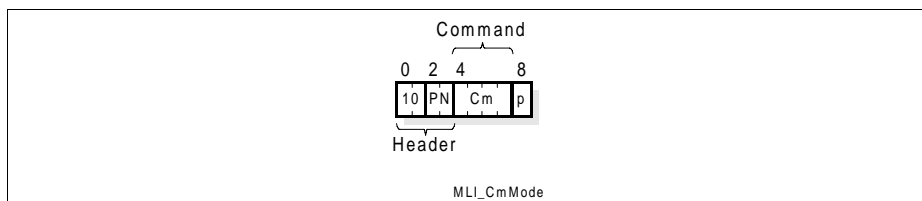


Figure 7-19 Command Frame

The bits denoted as “Cm” in the drawing above represent the command. Its value is obtained from the command bit field contained in the MLI transmitter command register (TCMDR.CMDPx, where x denotes the pipe number). The meaning of the command depends on the pipe used.

The command is a subset of four bits which encoding is shown in [Table 7-22 “Command Frame Encoding” on Page 7-38](#). The actions indicated are performed by the MLI receiver of the second microcontroller. In this table, DPE stands for delay for parity error (stored in [RCR.DPE](#)). It indicates the number of TCLK clock periods that the MLI receiver must wait without raising again the READY signal to inform that a parity error was detected.

[Table 7-6](#) illustrates where each of the fields of the frame are taken from.

Table 7-6 Storage of the Values Used in the Frame

Field	Value Taken From
PN	The correspondent to the accessed part of the command register
CMD	TCMDR.CMDPx

Note: x indicates the pipe number, $x = 0, 1, 2, 3$.

The number of bits transmitted will be as are described in [Table 7-7](#).

Micro Link Serial Bus Interface (MLI)

Table 7-7 Bits Transmitted in Command Frame

Header	Command	Parity	Total
4 bits	4 bits	1 bit	9 bits

CPU, DMA, or PCP writing within Transfer Window

- Write in Offset and Data Frame: the address offset can not be predicted.

Its frame code is 01_B. The MLI transmitter sends the offset and data frame. **Figure 7-20** illustrates this transfer. The number of bits of the address offset (m) and the bits of data (n) are known and specified in the transmitter status register of the current pipe.

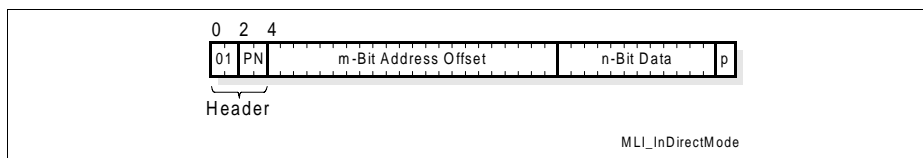


Figure 7-20 Write Access in Offset and Data Frame

Table 7-8 illustrates where each of the fields of the frame are taken from.

Table 7-8 Storage of the Values Used in the Frame

Field	Value Taken From
PN	The correspondent to the accessed registers
Address Offset	TPxAOFR
Data	TPxDATAR

Note: x indicates the pipe number, x = 0, 1, 2, 3.

The number of bits transmitted will be as shown in **Table 7-9**.

Table 7-9 Number of Bits In Offset And Data Frame

Data Width	Header	Offset	Data	Parity	Total
8 bits	4 bits	m bits	8 bits	1 bit	13+m bits
16 bits	4 bits	m bits	16 bits	1 bit	21+m bits
32 bits	4 bits	m bits	32 bits	1 bit	37+m bits

Micro Link Serial Bus Interface (MLI)

- Optimized Write Frame: the new address offset can be predicted.

Its frame code is 11_B. **Figure 7-21** shows this transmission mode.

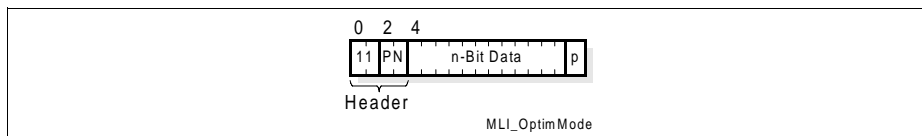


Figure 7-21 Optimized Write Frame

The offset will be deducted in the MLI receiver of the second microcontroller depending on the address prediction factor associated to the current pipe, TPxSTATR.AP in the transmitter side and RPxSTATR.AP in the receiver side.

Table 7-10 illustrates where each of the fields of the frame are taken from.

Table 7-10 Storage of the Values Used in the Frame

Field	Value Taken From
PN	The correspondent to the accessed registers
Data	TPxDATAR

Note: x indicates the pipe number, x = 0, 1, 2, 3.

The number of bits transmitted will be as shown in **Table 7-11**.

Table 7-11 Number of Bits In Optimized Write Frame

Data Width	Header	Data	Parity	Total
8 bits	4 bits	8 bits	1 bit	13 bits
16 bits	4 bits	16 bits	1 bit	21 bits
32 bits	4 bits	32 bits	1 bit	37 bits

Micro Link Serial Bus Interface (MLI)

CPU, DMA, or PCP Reading within Transfer Window

- Discrete Read Frame: the new offset address can not be predicted.

Its frame code is 01_B . The MLI transmitter sends the offset it wants to read from, indicating the width of the data. The frame in this case is shown in [Figure 7-22](#).

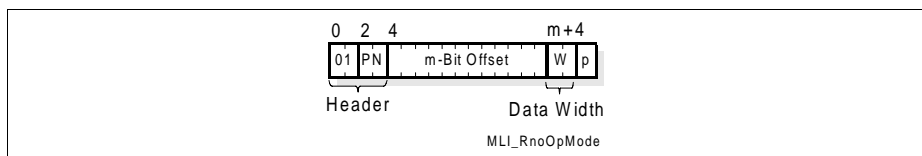


Figure 7-22 Discrete Read Frame

The field referenced as “W”, (data width) in the figure above will indicate the MLI receiver of the second microcontroller the size of data that this first microcontroller wants to read.

[Table 7-12](#) shows the encoding of this field.

Table 7-12 Data Width Encoding

W	Data Width
00_B	8 bits
01_B	16 bits
10_B	32 bits
11_B	Reserved

[Table 7-13](#) illustrates where each of the fields of the frame are taken from.

Table 7-13 Storage of the Values Used in the Frame

Field	Value Taken From
PN	The correspondent to the accessed registers
W	TPxSTATR.DW
Address Offset	TPxAOFR

Note: x indicates the pipe number, x = 0, 1, 2, 3.

Micro Link Serial Bus Interface (MLI)

The number of bits transmitted will be as shown in [Table 7-14](#).

Table 7-14 Number of Bits In Discrete Read Frame

Header	Data Width	Offset	Parity	Total
4 bits	2 bits	m bits	1 bit	7+m bits

Note: This case is perfectly distinguishable from the write in offset and data frame because for each pipe the value of the buffer size (TPxSTATR.BS = m, where x indicates the current pipe) is fixed all the time.

- Optimized Read Frame: the new address offset can be predicted.

Its frame code is 11_B. The frame in this case is shown in [Figure 7-23](#).

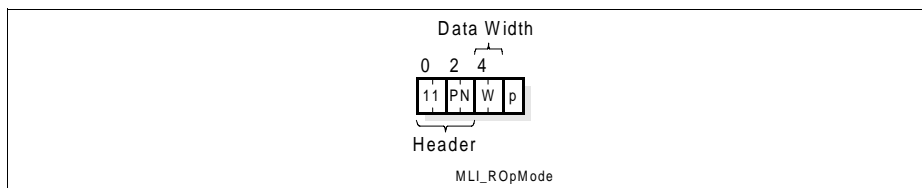


Figure 7-23 Optimized Read Frame

The bit field data width (W), has the same meaning as explained in [Table 7-12](#).

[Table 7-15](#) illustrates where each of the fields of the frame are taken from.

Table 7-15 Storage of the Values Used in the Frame

Field	Value Taken From
PN	The correspondent to the accessed registers
W	TPxSTATR.DW

Note: x indicates the pipe number, x = 0, 1, 2, 3.

The number of bits transmitted will be as shown in [Table 7-16](#).

Table 7-16 Number of Bits In Optimized Read Frame

Header	Data Width	Parity	Total
4 bits	2 bits	1 bit	7 bits

Micro Link Serial Bus Interface (MLI)

Answer Frame

Its frame code is 10_B. The MLI transmitter sends the frame illustrated in [Figure 7-24](#)

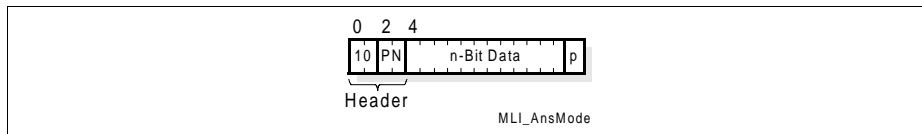


Figure 7-24 Answer Frame

The frame contains a data bit field that is the answer to a read operation made before. [Table 7-17](#) illustrates where each of the fields of the frame are taken from.

Table 7-17 Storage of the Values Used in the Frame

Field	Value Taken From
PN	TSTATR .APN
Data	TDRAR

An answer frame can be sent through any pipe which did receive a read request. There must be only one MLI transfer window read access pending on any side of a MLI connection at any time, because the answer mechanism does not contain buffers for multiple answer frames.

The number of bits transmitted will be as shown in [Table 7-18](#).

Table 7-18 Number of Bits In Answer Frame

Data	Header	Parity	Total
8 bit	4 bit	1 bit	13 bit
16 bit	4 bit	1 bit	21 bit
32 bit	4 bit	1 bit	37 bit

7.2.6.6 Transfer Mode Selection

Each time a write access to the MLI registers is made a related flag is set in the transmitter registers status register (**TRSTATR**), indicating that the data stored in them has not been sent (transfer pending) and no more writing accesses may be made on these registers. These flags are reset again when the registers may be again written.

Table 7-19 shows the flags that are set when accessing the different registers and when they are reset again.

Table 7-19 Valid Flags

Flag	Set When Access to	Reset When
TRSTATR.DVx	TPxAOFR, TPxDATAR for read or write operation	The write or read frame has been sent correctly through the correspondent pipe
TRSTATR.RPx	TPxAOFR, TPxDATAR for read operation	The answer frame has been received correctly through the correspondent pipe
TRSTATR.CIVx	The MLI transmitter detects a rising edge in the trigger_commandx line	The correspondent command frame has been sent correctly through the pipe 0
TRSTATR.CVx	TCMDR or CVx is set via software	The command frame has been sent correctly through the correspondent pipe
TRSTATR.BAV	TCBAR	The copy base address frame has been sent correctly through the correspondent pipe
TRSTATR.AV	TDRAR	The answer frame has been sent correctly through the correspondent pipe

Note: x indicates the pipe number, x = 0, 1, 2, 3.

When a rising edge is detected in the trigger_commandx line, the MLI transmitter will set the correspondent CIVx bit, and it will send a command frame through pipe 0 as is explained in **Table 7-20**.

Table 7-20 Hardware Triggered Command

Rising Edge	Command to Send Through Pipe 0
Trigger_command0	0001 _B
Trigger_command1	0010 _B
Trigger_command2	0011 _B
Trigger_command3	0100 _B

When more than one transfer was pending the criteria to choose between the different frames to send will be based in a priority scheme as follows:

- Read answer frame, command frame triggered by hardware, command frame triggered by software, read frame, write frame and copy base address frame.
- If in two or more pipes there are operations pending with the same priority, then it is chosen the one from the pipe with highest priority. The pipe number 0 has the highest priority, and then pipe 1, 2 and 3 by this order.

The following paragraphs illustrate the different actuations taken depending on the type of frame chosen to be sent:

Copy Base Address Frame

The **TRSTATR**.BAV flag is set to one, and there is not other type of frame pending in this pipe. This transfer means that the microcontroller wants to initialize the corresponding pipe base address of the MLI receiver in the other microcontroller and its correspondent buffer size. The MLI transmitter will send the copy base address frame with the base address stored in the TCBAR register and the buffer size stored in the correspondent TPxBAR register.

After the transfer of a base address frame, the next two data frames are not in optimized mode in order to ensure a correct offset prediction.

Write and Read Frame

- If only the **TRSTATR**.DVx flag is set, it indicates a write operation.
- If **TRSTATR**.DVx and **TRSTATR**.RPx flags are set, they indicate a read operation.

Depending on the operation code this will mean that the first microcontroller wants to write the data (stored in TPxDATAR) in the relative address indicated by the address offset (that will be stored in TPxAOFR) of the transfer window defined by the selected pipe or that it wants to read data from that position.

If the address prediction method is allowed (**TCR**.NO = 0, where x indicates the pipe) then the MLI transmitter will compare the new address offset written in the bus with the old one (TPxAOFR in the moment of accessing). The difference between these two

Micro Link Serial Bus Interface (MLI)

addresses is stored in the correspondent pipe status register (TPxSTATR.AP) if it is not greater than 10 bits (in two's complement). If the difference is the same in two consecutive transfers then the MLI transmitter will set the optimized flag in the status register of the correspondent pipe (TPxSTATR.OP). This will indicate that the optimized mode will be used to send this frame when it was chosen to be sent. The optimized mode will be used for the rest of the frames, as far as TPxSTATR.OP = 1, until the difference was not the same as the one stored in the status register, (then the MLI resets again TPxSTATR.OP).

When the conditions explained above are not met, then the transfer is performed using a normal frame for writing or reading (discrete read frame or write in address offset and data frame).

When the necessary comparisons to infer the address prediction factor, the new address offset is finally stored in the address offset register of the pipe (TPxAOFR).

Next pseudocode illustrates the process:

```

if (TPxSTATR.NO = 0) then // Optimized mode possible
    delta = new_address_offset - TPxAOFR
    if (delta = TPxSTATR.AP) then
        if (Coincidence = TRUE) then
            TPxSTAT.OP = 1
        else
            Coincidence = TRUE
            TPxSTAT.OP = 0
        end if
    else if (if difference delta is not bigger than 9 bits) then
        TPxSTATR.AP = delta
        Coincidence = FALSE
        TPxSTAT.OP = 0
    else
        Coincidence = FALSE
        TPxSTAT.OP = 0
    end if
else // Optimized mode disabled
    TPxSTAT.OP = 0
end if

```

Note: TPxAOFR contains the last address offset used in the pipe. TPxSTATR.AP is the address prediction factor. The boolean variable Coincident, expresses the condition of two consecutive coincidences in the address offsets.

When the data of the pipe was chosen to be sent, using the priority method before explained, the MLI will know if the frame should be or not sent using the optimized mode. If TPxSTATR.OP the optimized mode must be used.

Micro Link Serial Bus Interface (MLI)

When the write or read frame is correctly received by the other microcontroller, the MLI resets the correspondent **TRSTATR.DVx** flag. In the case of a read operation, the flag **TRSTATR.RPx** will be reset by hardware when the answer was received. Until this read pending flag is not reset again, the registers TPxDATAR and TPxAOFR of the pipe may not be written again.

Answer Data Frame

This will mean that this written data in the **TDRAR** register is the answer to a read operation so that the transmission will be made in answer frame.

When the writing frame is correctly received by the other microcontroller, the MLI resets the **TRSTATR.AV** flag.

Command Frame

If CIVx is set, the MLI transmitter will send the command frame correspondent to the line through the pipe 0 as explain in **Table 7-20**. When the command frame is correctly received by the other microcontroller, the MLI resets the correspondent **TRSTATR.CIVx** flag.

If CVx is set, the MLI transmitter will check the value stored in the command bit field correspondent to the pipe of the TCMR register and it will send it using the command frame. When the command frame is correctly received by the other microcontroller, the MLI resets the correspondent **TRSTATR.CVx** flag.

7.2.6.7 Parity Generation

The type of parity used will be even or odd parity, depending on the programmed value in **TCR.TP**. The parity bit will be calculated by toggling a bit each time a one is sent in the frame. The starting value for the toggling bit will be zero if the parity is even, and one if the parity is odd. Assuming a correct transmission, a starting value of 1 in the toggle bit will lead into a parity error situation as explained in **Section 7.2.4**.

Note: There is no protection against frames having more than one corrupted bit (e.g. shortened frames). In such cases, an unpredictable behavior of the MLI module may occur.

7.2.6.8 Error Detection and Handling

The MLI transmitter will be able to recognize the following error situations in the transmission:

- A non acknowledged transfer
- A parity error

Non Acknowledge Error Detection

This situation is as explained in [Figure 7-12](#) and in [Figure 7-40](#). When this error is detected, the MLI transmitter will set the non acknowledge error flag in its control register (**TSTATR.NAE**) and decreases the counter of non acknowledge errors (**TCR.MNAE**). When this counter reaches the value zero, a time out interrupt is generated if enabled.

Parity Error Detection

The Parity Error situation is as explained in [Figure 7-13](#) and in [Figure 7-39](#). When this error is detected, the MLI transmitter will set the parity error flag in its control register (**TSTATR.PE**) and decreases the counter of parity errors (**TCR.MPE**). When this counter reaches the value zero, a parity error interrupt is generated if enabled.

Whenever the MLI transmitter sends correctly a new frame, it will produce an interrupt, if enabled, depending on the type of frame: command frame interrupt or normal frame interrupt, see [Section 7.2.9](#).

7.2.6.9 MLI Transmitter Input/Output Control

Figure 7-25 shows the control structure for the transmitter output signals VALID, CLK and DATA. The VALID signal can be distributed to up to four output lines (TVALIDA to TVALIDD). It is possible to individually enable/disable each line (except DATA) and to select its polarity.

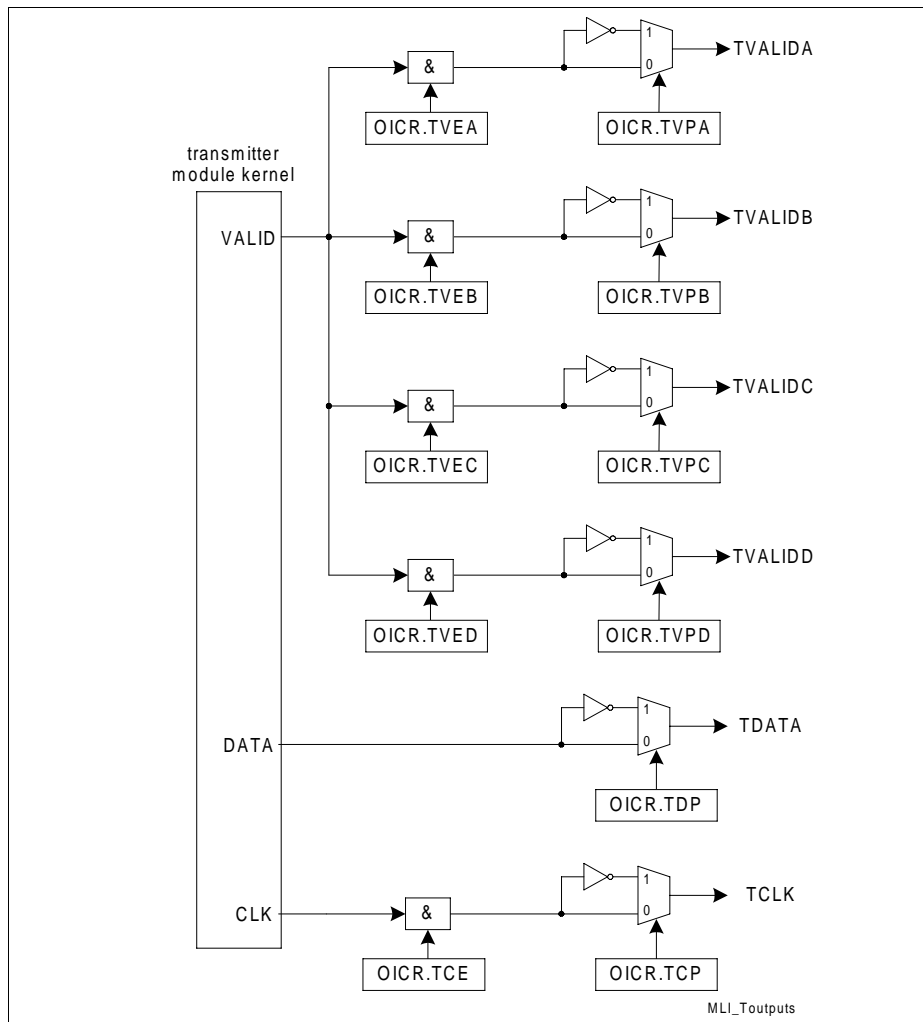


Figure 7-25 Control of the Transmitter Output Signals

Micro Link Serial Bus Interface (MLI)

The transmitter output shift clock signal CLK can be enabled by the bit **OICR.TCE** and its output polarity can be selected by **OICR.TCP**. For the data signal DATA it is possible to select the polarity by programming bit **OICR.TDP**.

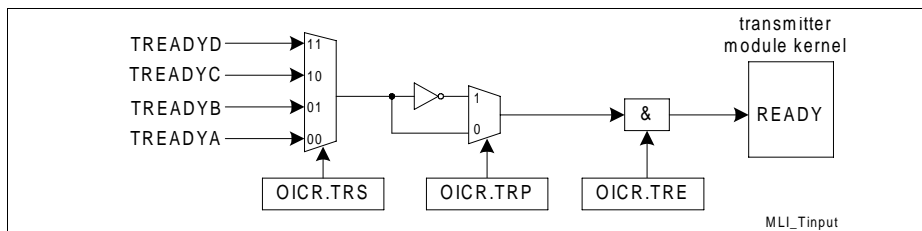


Figure 7-26 Control of the Transmitter Input Signal

The transmitter input signal READY can be selected from four possible input signals (TREADYA to TREADYD). The selected input signal can be enabled by bit **OICR.TRE** and its polarity can be chosen by **OICR.TRP**.

Note: The first letter 'T' of the signal names indicates that these signals are belonging to the transceiver part of an MLI module. The last letter 'A' to 'D' of a signal belonging to a set of lines indicates that the signal can be selected from (input) or can be distributed to (output) up to 4 lines.

7.2.7 MLI Receiver

7.2.7.1 MLI Receiver Reset

After the hardware reset the MLI receiver will be in receiver off mode as it will be explained in [Section 7.2.7.2](#).

7.2.7.2 MLI Receiver Operation Modes

By programming the MLI receiver control register it is possible to set its operation mode. In [Table 7-21](#) are shown all the possible modes of the MLI receiver, depending on the values of the mode parameter (**RCR.MOD**). The characteristics of the different modes are explained below.

Table 7-21 MLI Receiver Operation Modes

RCR.MOD	Mode of Operation
0	MLI move engine off, receiver off
1	MLI move engine in automatic mode

Micro Link Serial Bus Interface (MLI)**Move Engine Off**

- The Ready signal is activated and the acknowledge and parity error conditions are accomplished.
- All the write or read transfers have no active effect on the FPI bus in the MLI receiver side.
- Modification of the control parameters or the base addresses of the transmission pipes are allowed.
- Commands are taken into account. Possibility of interrupts programming.

Move Engine in Automatic Mode

Its characteristics are the following ones:

- The Ready signal is activated and the acknowledge and parity error conditions are accomplished.
- All the data transfers are taken into account.
- Modification of the control parameters or the base addresses of the transmission pipes are allowed.

This bit may be modified by the MLI receiver whenever it receives the proper command.

Note: The next paragraphs will explain the whole MLI receiver functionality and its interfaces arbitration for the Receiver On operation mode. In every case must be taken into account the operation mode in which the MLI receiver is programmed and consider the limitations introduced by each of the explained modes.

Micro Link Serial Bus Interface (MLI)

7.2.7.3 Internal Architecture And Interface Signals

Figure 7-27 shows the internal architecture of the MLI receiver:

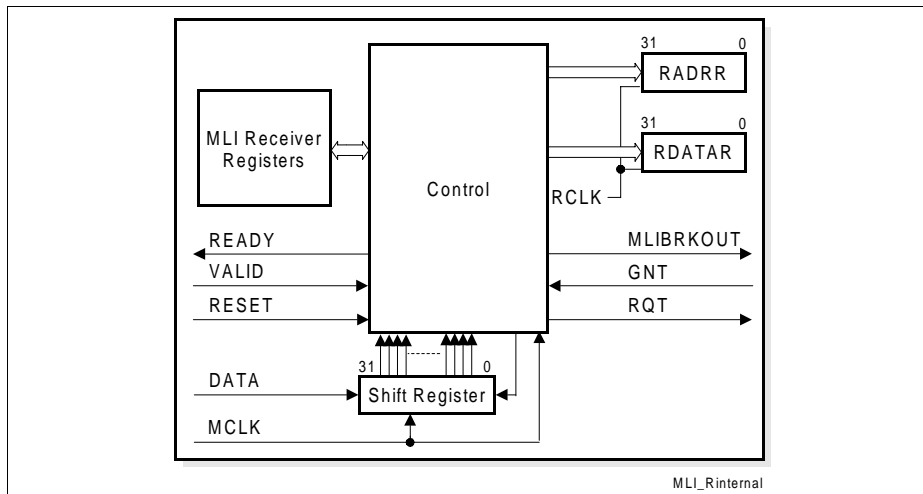


Figure 7-27 MLI Receiver Internal Architecture

Note: The letter “x” indicates the pipe number, from 0 to 3.

The signal RQT is used to notify the DMA that the MLI receiver has new data in its registers **RADRR** and **RDATAR**. When the DMA reads the registers **RADRR** and **RDATAR**, the MLI interface sets high the signal gnt. From both MLI interface point of view the signals are as explained in [Section 7.2.3](#).

Each one of the registers RPxBAR (where x indicates the pipe) will contain a default value for the base address of each of the four pipes. Each time the MLI receiver obtains a new frame it will recognize the kind of transmission and the pipe from the header and from the number of received bits. The MLI receiver will keep track of the next parameters:

- Width of the current data received in the pipe (RPxSTATR.DW, where x denotes the pipe number)
- Width of each transfer window (RPxSTATR.BS, where x denotes the pipe number)

A complete list of the MLI receiver registers may be found in [Section 7.3.3](#).

7.2.7.4 MLI Receiver Operation

The MLI receiver will obtain the information sent from the MLI transmitter from the first microcontroller. It will proceed to operate over the information received in order to extract each of the bit fields. The final step is that the data and or address registers (**RDATAR**,

Micro Link Serial Bus Interface (MLI)

RADRR) are updated with a new value. Due to the fact that all the data is received synchronized with the clock of the MLI transmitter from the other microcontroller, the MLI receiver must synchronize the address and the data values with its internal clock (RCLK) and the registers **RDATAR** and **RADRR** will contain their values.

Its operation will depend directly on the type of frame received. A complete reference of the different modes may be found in **Section 7.2.6.5**.

In order to accomplish the information split, the MLI receiver will have into account the number of bits of the received frame, the transmission mode and the buffer size for the current pipe.

Copy Base Address Frame

Its description and the number of bits of this frame may be seen in **Figure 7-18** and in **Table 7-5** respectively.

After the header, the MLI receiver obtains the 28 MSB's of the current pipe base address that will be stored in the 28 MSB's of its corresponding pipe base address register (RPxBAR), the buffer size is stored in the status register (RPxSTATR.BS), and the Type of frame bit field is updated (**RCCR.TF** = 00_B). A normal frame received interrupt is produced if enabled by **RIER.NFRIE**.

Micro Link Serial Bus Interface (MLI)

Command Frame

Its description and the number of bits of this frame may be seen in [Figure 7-19](#) and in [Table 7-7](#) respectively. After the complete reception of this frame the MLI receiver splits the information in its different bit fields.

[Table 7-22](#) illustrates the different values that may be received in the command frame and the action taken for each one.

Table 7-22 Command Frame Encoding

PN	Cm	Action
00 _B	0001 _B	Generate interrupt 0, if enabled by RIER .ICE
	0010 _B	Generate interrupt 1, if enabled by RIER .ICE
	0011 _B	Generate interrupt 2, if enabled by RIER .ICE
	0100 _B	Generate interrupt 3, if enabled by RIER .ICE
	Others	No effect
01 _B	0000 _B	Set RCR .DPE = 0000 _B
	0001 _B	Set RCR .DPE = 0001 _B
	0010 _B	Set RCR .DPE = 0010 _B
	0011 _B	Set RCR .DPE = 0011 _B

10 _B	1111 _B	Set RCR .DPE = 1111 _B
	0001 _B	Set RCR .MOD = 1, enable automatic mode
	0010 _B	Set RCR .MOD = 0, disable automatic mode (listen mode)
	0100 _B	Reset TRSTATR .RP0
	0101 _B	Reset TRSTATR .RP1
	0110 _B	Reset TRSTATR .RP2
	0111 _B	Reset TRSTATR .RP3
	1111 _B	Generate a pulse on line <u>MLIBRKOUT</u> (if enabled by RCR .BEN)
11 _B	others	No effect
	Any	Command meaning interpreted by software

Micro Link Serial Bus Interface (MLI)

When the MLI receiver gets through the pipe 2 ($PN = 10_B$) the command 1111_B then it asserts the signal MLIBRKOUT, active in low-level if enabled by RCR.BEN bit. **Figure 7-28** illustrates this procedure.

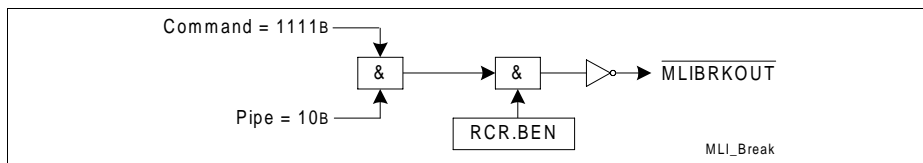


Figure 7-28 Assertion of MLIBRKOUT Signal

Write Access In Offset And Data Frame

Its description and the number of bits of this frame may be seen in **Figure 7-20** and in **Table 7-9** respectively.

After the header, the MLI receiver obtains the m bits corresponding to the offset. The MLI receiver will know how many bits correspond to the address offset because is the same number as indicated in the buffer size for the current pipe (RPxSTATR.BS).

In order to follow the same address prediction method that is carried out by the MLI transmitter, the MLI receiver will compare the address offset of the currently received frame with the address offset previously received. This last address offset is in the receiver base address register of the pipe (the RPxSTATR.BS LSB's of RPxBAR, where $x = 0, 1, 2, 3$ indicates the pipe). If the difference between both addresses is less than 9 bits, the MLI transmitter will proceed to store it in the address prediction factor bit field (RPxSTATR.AP), and this value will be used to obtain the address whenever an optimized frame was received. After this comparison, the newly received address offset is stored in the lowest part of RPxBAR. **Figure 7-29** illustrates this process.

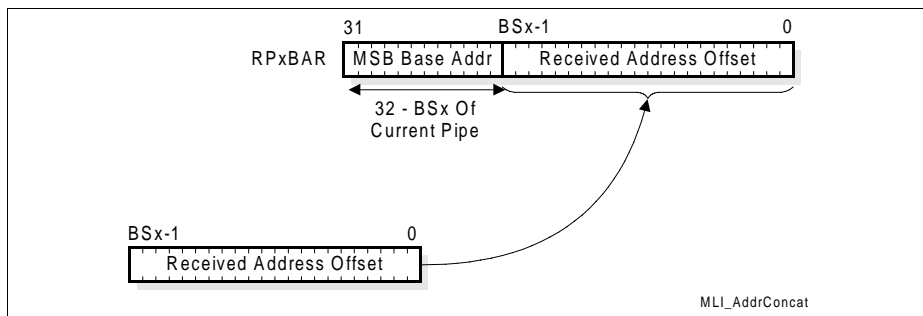


Figure 7-29 Absolute Address Obtaining

Note: BS stands for buffer size (in bits) of the current pipe (x).

Micro Link Serial Bus Interface (MLI)

After the concatenation, this absolute address will be stored in the MLI receiver address register **RADDR**. The next subset of bits will be corresponding to the data and the parity bit. This bits will be stored in the **RDATAR**. The **RCR.DW** and **RCR.TF** bit fields are updated, **RCR.TF** = 10. A normal frame received interrupt is produced if enabled by **RIER.NFRIE**.

Table 7-23 shows the place of storage of the information obtained from the frame.

Table 7-23 Place of Storage For Address Offset And Data Bits

m-Bit Address Offset	n-bit Data
BS least significant bits of RPxBAR. After that, copy the 32 bits of the register RPxBAR in RADDR	RDATAR

Optimized Write Frame

Its description and the number of bits of this frame may be seen in **Figure 7-21** and in **Table 7-11** respectively.

After the header, the MLI receiver obtains n bits of data that will be stored in the MLI receiver data register (**RDATAR**).

The address offset must be calculated by adding to the last address offset used (the content of RPxBAR register) the address prediction factor (RPxSTATR.AP, 10 bits with one sign bit).

Figure 7-30 illustrates this address offset process obtaining.

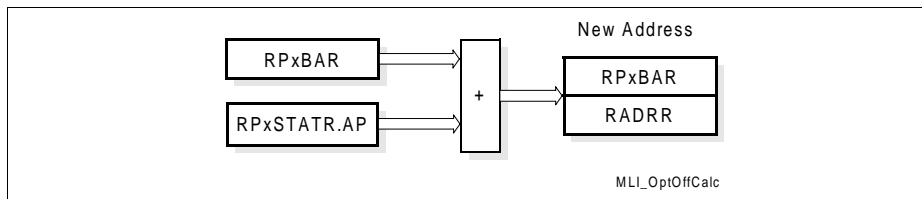


Figure 7-30 Address Offset Prediction for Optimized Write Frame

If the increment or decrement of the address offset results in over or underflow, then the wraparound method is used to fit the new address to the transfer window.

This newly obtained 32 bits address is stored in RPxBAR and in **RADDR** registers. The **RCR.DW** and **RCR.TF** bit fields are updated, **RCR.TF** = 10. A normal frame received interrupt is produced if enabled by **RIER.NFRIE**.

Micro Link Serial Bus Interface (MLI)

Discrete Read Frame

Its description and the number of bits of this frame may be seen in [Figure 7-22](#) and in [Table 7-14](#) respectively.

After the header, the MLI receiver obtains two bits that indicates the width of the data that must be read. The next m bits (as much as $RPxSTATR.BS$) represent the offset from which this data will be read from, and the parity bit.

In order to follow the same address prediction method that is carried out by the MLI transmitter, the MLI receiver will compare the address offset of the currently received frame with the address offset previously received. This last address offset is in the receiver base address register of the pipe (the $RPxSTATR.BS$ LSB's of $RPxBAR$, where $x = 0, 1, 2, 3$ indicates the pipe). If the difference between both addresses is less than 9 bits, the MLI transmitter will proceed to store it in the address prediction bit field ($RPxSTATR.AP$), and this value will be used whenever an optimized frame was received.

The absolute address is calculated by concatenating the obtained address offset with the base address of the current pipe as explained in [Figure 7-29](#) and then it is stored in the register **RADRR**. The **RCR.DW** and **RCR.TF** bit fields are updated, **RCR.TF** = 01. A normal frame received interrupt is produced if enabled by **RIER.NFRIE**.

[Table 7-24](#) shows the place in which each of the received bit fields are stored.

Table 7-24 Place of Storage

Data Width	RPxSTATR.BS bits of Address Offset
RPxSTATR.DW	Concatenate with the (32 - RPxSTATR.BS) bits more significant of RPxBAR . Copy this new value of RPxBAR in RADRR .

Optimized Read Frame

Its description and the number of bits of this frame may be seen in [Figure 7-23](#) and in [Table 7-16](#) respectively.

After the header, the MLI receiver obtains two bits that indicates the width of the data that must be read.

The absolute address must be calculated by adding to the last address used (**RPxBAR**) the address prediction factor (**RPxSTATR.AP**, 10 bits with one sign bit) as explained in [Figure 7-30](#). If the increment or decrement of the address offset results in over or underflow, then the wraparound method is used to fit the new address to the transfer window.

The 32 bits resulting absolute address will be stored in the **RADRR** and **RPxBAR** registers. The **RCR.DW** and **RCR.TF** bit fields are updated, **RCR.TF** = 10. A normal frame received interrupt is produced if enabled by **RIER.NFRIE**.

Answer Frame

Its description and the number of bits of this frame may be seen in [Figure 7-24](#) and in [Table 7-18](#) respectively. If an answer frame is received and **TRSTATR.RPx** = 1 and **TRSTATR.DVx** = 0, where x is the pipe from which the frame was received, then the frame is discarded and a discarded read answer interrupt is produced if enabled by **RIER.DRAIE**.

After the header, the MLI receiver obtains n bits of data that will be stored in the **RDATAR** register. This data is the answer to a read operation started by the MLI transmitter of the same MLI. The **RCR.DW** and **RCR.TF** bit fields are updated, **RCR.TF** = 11.

The MLI receiver will then produce a normal frame received interrupt controlled by **RIER.NFRIE**.

7.2.7.5 Error Handling

The parity bit is checked using the same method as explained in [Section 7.2.6.7](#). If the MLI receiver detects a parity error in the transmission it will raise the READY signal following the next conditions:

- If the MLI receiver is prepared to receive a new frame before the delay to indicate parity error programmable value (**RCR.DPE**) clock periods, then it will wait until **RCR.DPE** clock periods had passed. This DPE value will be received with a command frame from the MLI transmitter in the other microcontroller.
- If by the contrary the MLI is not prepared before **RCR.DPE** clock periods had passed, then it will raise the READY signal as soon as it was prepared to receive a new frame.

This will inform the transmitter in the other microcontroller that a parity error was detected. If the parity error is signaled, the MLI receiver sets the parity error flag (**RCR.PE**) and decreases the counter of parity errors (**RCR.MPE**). When this counter reaches the value zero, a parity error interrupt is generated if enabled.

The way the MLI receiver in the second frame informs the MLI transmitter a parity error situation is illustrated in [Figure 7-31](#).

Micro Link Serial Bus Interface (MLI)

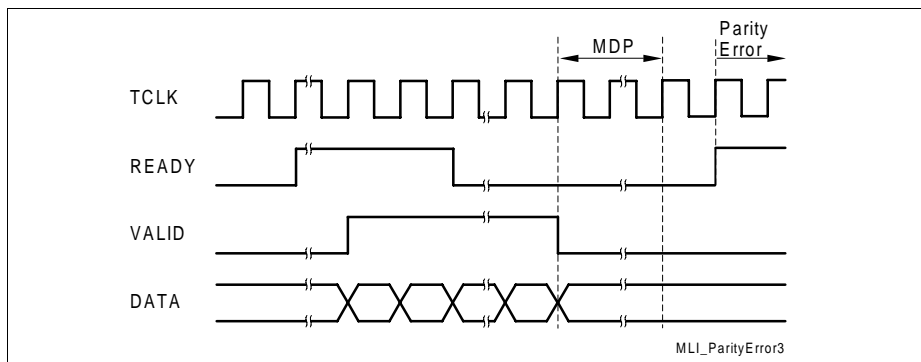


Figure 7-31 Parity Error

*Note: The signals are seen from the MLI transmitter side. When VALID is not asserted the DATA line will have only a value (one or zero) depending on the programmed value in **TCR.DNT** and its chosen polarity.*

Micro Link Serial Bus Interface (MLI)

7.2.7.6 MLI Receiver Input/Output Control

Figure 7-32 shows the control structure for the receiver input signals VALID, CLK and DATA. Each input signal to the receiver module kernel can be selected from up to four input lines. It is possible to individually enable/disable each line (except DATA) and to select its polarity.

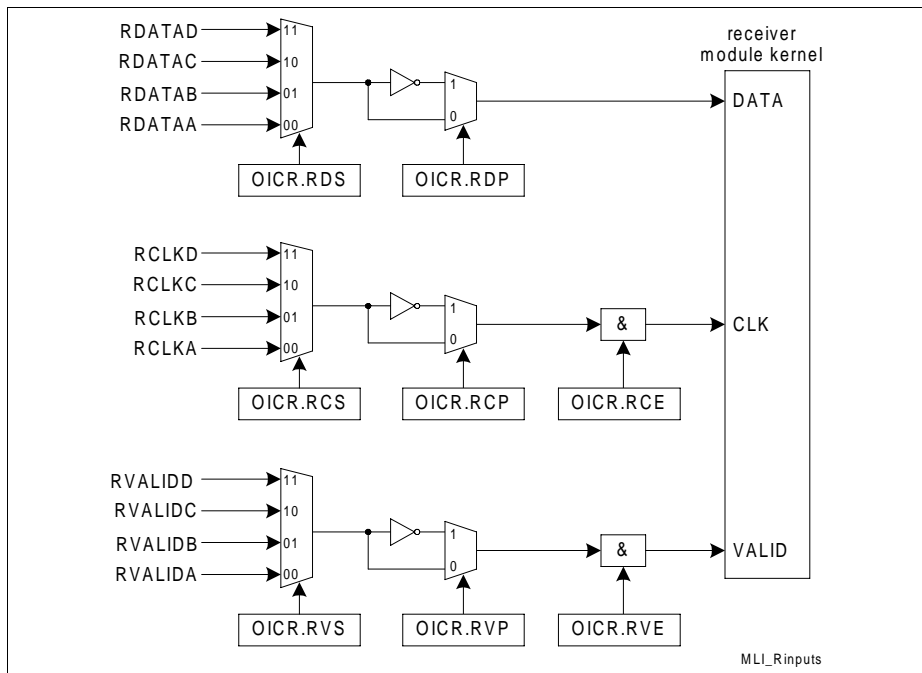


Figure 7-32 Control of Receiver Input Signals

Note: The first letter 'R' of the signal names indicates that these signals are belonging to the receiver part of an MLI module. The last letter 'A' to 'D' of a signal belonging to a set of lines indicates that the signal can be selected from (input) or can be distributed to (output) up to 4 lines.

The receiver output signal READY can be distributed to up to four output lines (RREADYA to READYD). It is possible to individually enable/disable each line and to select its polarity. see **Figure 7-33**.

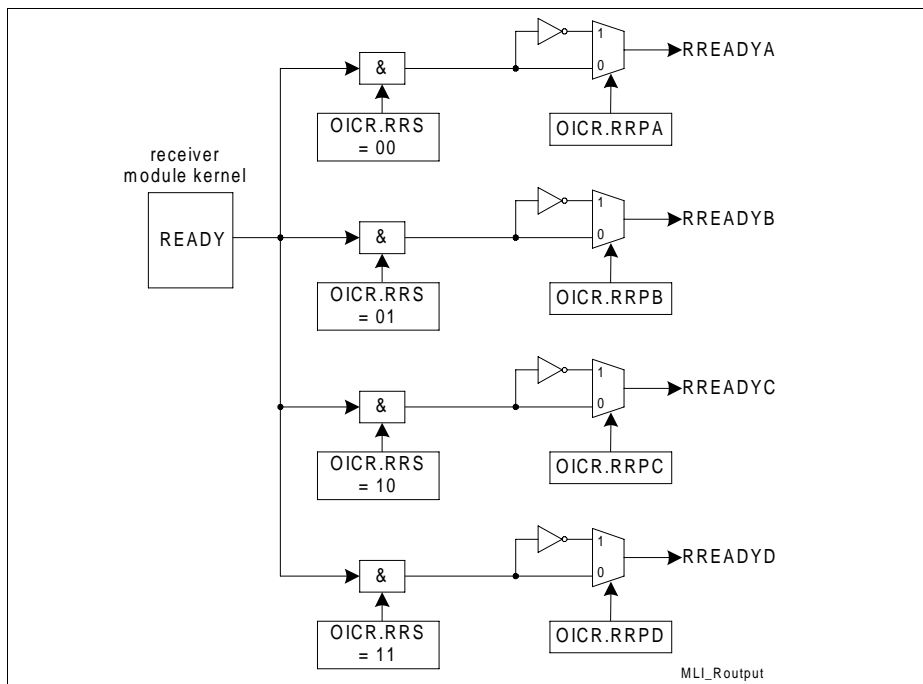


Figure 7-33 Control of Receiver Output Signal

For the CLK and DATA lines it is possible to choose which of them will be active with the **OICR.RCS** and **OICR.RDS** bit fields. With the same purpose as before, for both outputs it is possible to set their polarities by programming the bits **OICR.RCP** and **OICR.RDP**.

Micro Link Serial Bus Interface (MLI)

7.2.8 Reading Process Summary

Figure 7-34 illustrates the whole reading process, since it is originated by the first microcontroller, until it gets the answer.

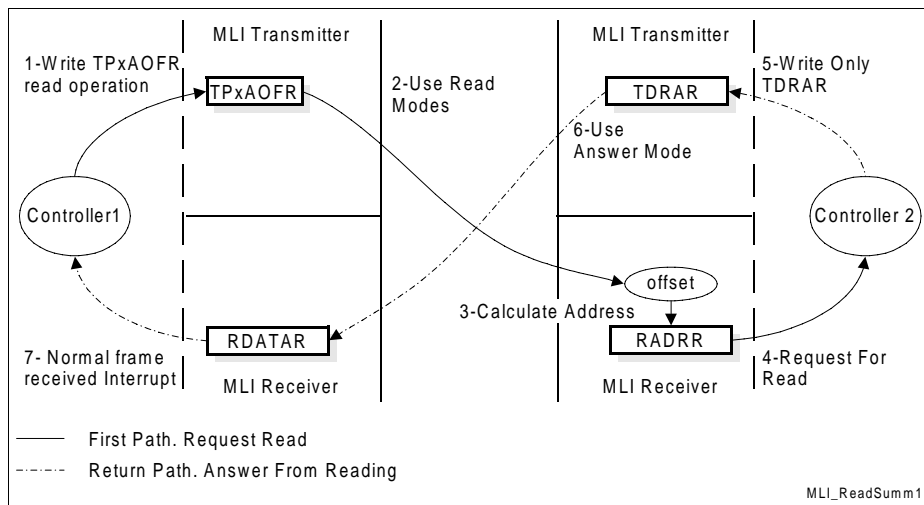


Figure 7-34 Full Read Operation Process

In the second step the MLI transmitter of the MLI in the microcontroller 1, sets **TRSTATR.RPx** and **TRSTATR.DVx** equal to one. Then it sends the read frame (optimized or not). When this frame is properly received, the MLI transmitter resets again the **TRSTATR.DVx** flag.

In step 3, the MLI receiver from the second microcontroller calculates the address and writes **TSTATR.APN** with the value received in the read frame. In 6, the MLI transmitter from the second microcontroller sends an answer frame using the pipe number indicated in **TSTATR.APN**.

In step number 7, the MLI receiver of the microcontroller 1 gets an answer frame. If a read pending flag is 0 and its DVx is 1, then the frame is discarded and the MLI receiver produces an interrupt if enabled by **RIER.DRAIE**. If the RPx is set and DVx reset, the MLI receiver produces a normal frame received interrupt (if enabled by **CIR.NFRIE**) to inform its CPU that it has the answer.

The software may read the data width and the address offset from the **TPxSTATR** **TPxAOFR** registers (the pipe is indicated by **RCR.PN**). After this read operation has been finished, a next read to the transfer window can be initiated.

Micro Link Serial Bus Interface (MLI)

7.2.9 MLI Interrupts

The general interrupt structure is shown in the figure below. The interrupt event can trigger the interrupt generation and sets the corresponding bit in the status register. The interrupt pulse is generated independently from the interrupt flag in the interrupt status register. The interrupt flag can be reset by SW.

If enabled by the related interrupt enable bit in the interrupt enable register, an interrupt pulse can be generated at one of the interrupt output lines INT_Ox of the module. If more than one interrupt source is connected to the same interrupt node pointer (in the interrupt node pointer register), the requests are combined to one common line.

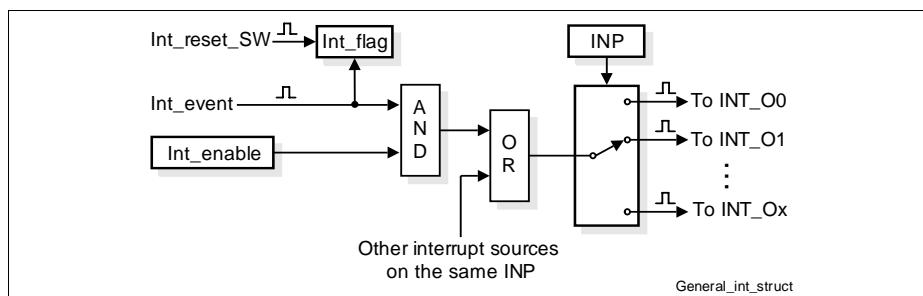


Figure 7-35 General Interrupt Structure

The request compressor condenses the MLI interrupt request sources to 8 interrupt outputs, reporting the interrupt requests of the MLI module to the interrupt microcontroller. Each request source is provided with an interrupt output pointer, selecting the interrupt output to start the associated service routine to increase flexibility in interrupt processing. Each of the 8 interrupt outputs can trigger an independent routine with its own interrupt vector and its own priority.

Micro Link Serial Bus Interface (MLI)

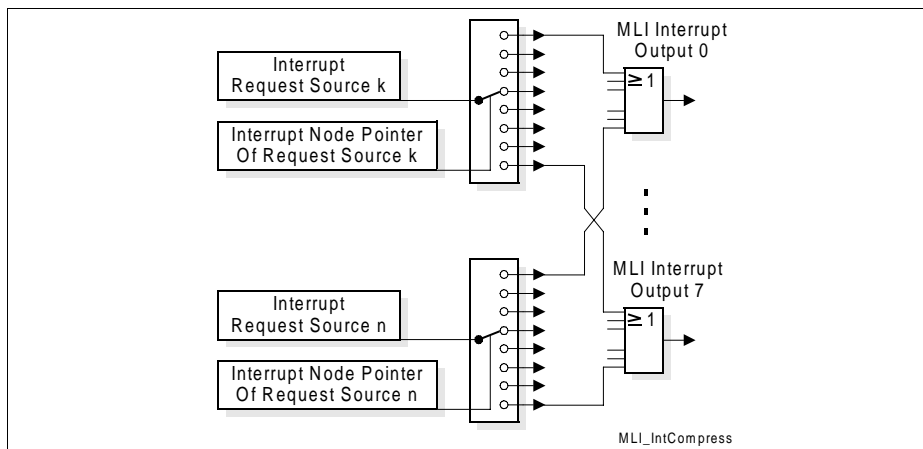


Figure 7-36 Interrupt Output Pointer and Interrupt Request Compressor

Micro Link Serial Bus Interface (MLI)

Table 7-25 shows the MLI interrupts.

Table 7-25 MLI Interrupts

Description and Condition	Agent that Generates it	Controlled by
Parity error interrupt. Produced when a programmable maximum number of parity errors is reached.	MLI transmitter	TINPR .PTEIP TIER .PEIE TIER .PEIR
Time out error interrupt. Generated when a programmable maximum number of non acknowledge errors is reached.	MLI transmitter	TINPR .PTEIP TIER .TEIE TIER .TEIR
Normal frame sent interrupt. MLI transmitter has sent a normal frame (not command) through pipe number x.	MLI transmitter	TINPR .NFSIPx TIER .NFSIEx TIER .NFSIRx
Command frame sent interrupt. MLI transmitter has sent a command frame through pipe number x, (only pipes 0, 1, 2 and 3).	MLI transmitter	TINPR .CFSIP TIER .CFSIEx TIER .CFSIRx
Discarded read answer received interrupt. Produced whenever an answer frame is received and the RPx flag of its correspondent pipe is 0, or RPx and DVx are 1.	MLI receiver	RINPR .DRAIP RIER .DRAIE RIER .DRAIR
Parity error interrupt. Produced when a programmable maximum number of parity errors is reached.	MLI receiver	RINPR .MPPEIP RIER .PEIR
Normal frame received interrupt. The MLI receiver has obtained a normal frame (not command).	MLI receiver	RINPR .NFRIP RIER .NFRIE RIER .NFRIR
Command frame received interrupt. The MLI receiver has obtained a command frame through pipe number x.	MLI receiver	RINPR .CFRIP RIER .CFRIEx RIER .CFRIRx

7.2.10 Clock Domains and Handshake Timing

Figure 7-37 illustrates how the signals are synchronized in the MLI transmitter and in the receiver. In the figure, the suffixes T1 and R2 indicate from which side the signals are seen (from the MLI1 transmitter or from the MLI2 receiver).

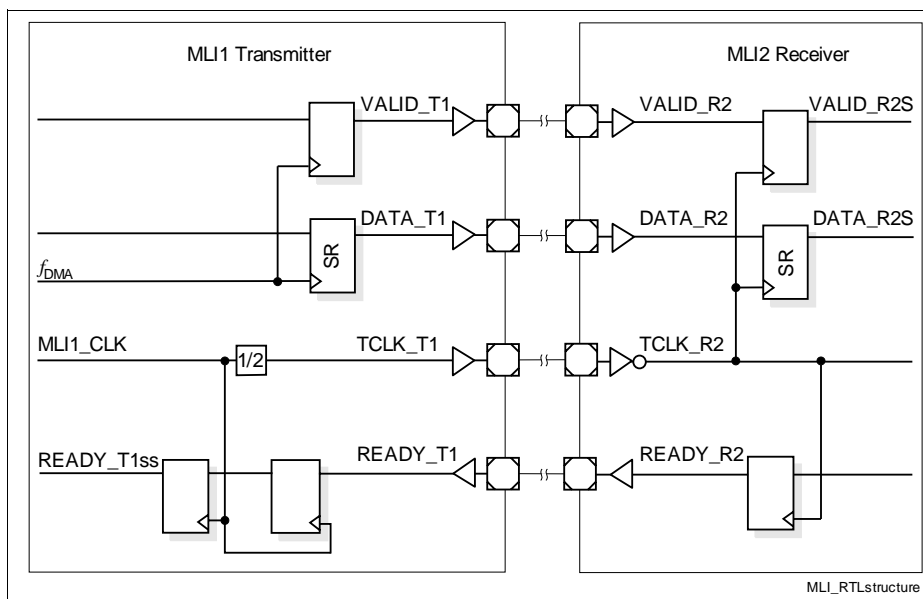


Figure 7-37 Signals Synchronization in MLI Transmitter and Receiver

Note: In the figure above SR stands for shift register.

The physical connection between both MLI will introduce a delay that will be dependent of the concrete application and must be characterized in order to choose proper values for MDP and RCR.DPE.

Figure 7-38 shows a detailed time diagram of a correct transfer. Each of the signals is as shown in **Figure 7-37**. The delay d1 is the one that affects the signals coming out from the MLI transmitter and d2 is the delay associated to the READY signal.

Micro Link Serial Bus Interface (MLI)

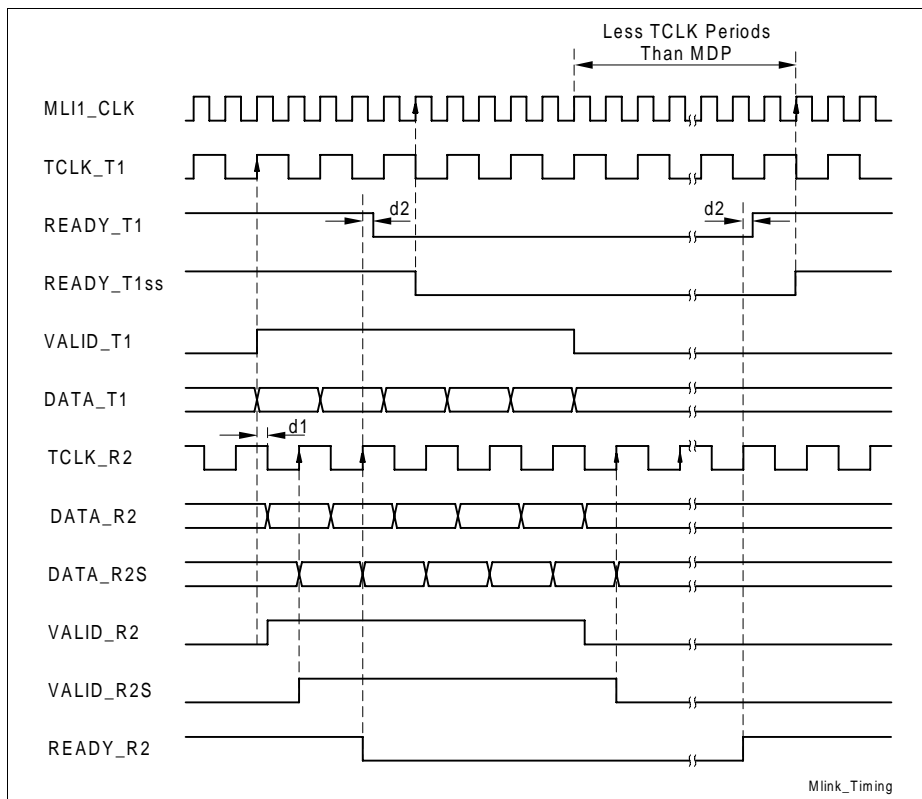


Figure 7-38 Detailed Handshake in a Correct Transfer

Figure 7-39 illustrates how the MLI receiver informs the transmitter that it has received a frame with a parity error.

Micro Link Serial Bus Interface (MLI)

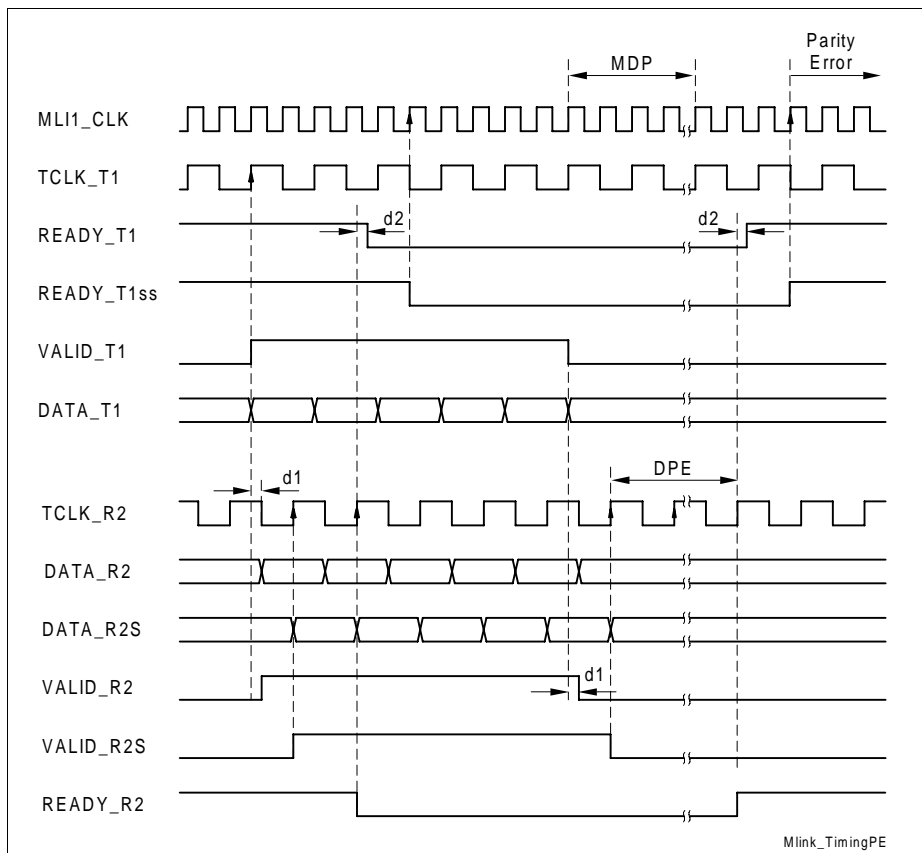


Figure 7-39 Detailed Handshake in a Transfer with Parity Error

Figure 7-40 illustrates the situation in which the MLI receiver has not acknowledged a transmission.

Micro Link Serial Bus Interface (MLI)

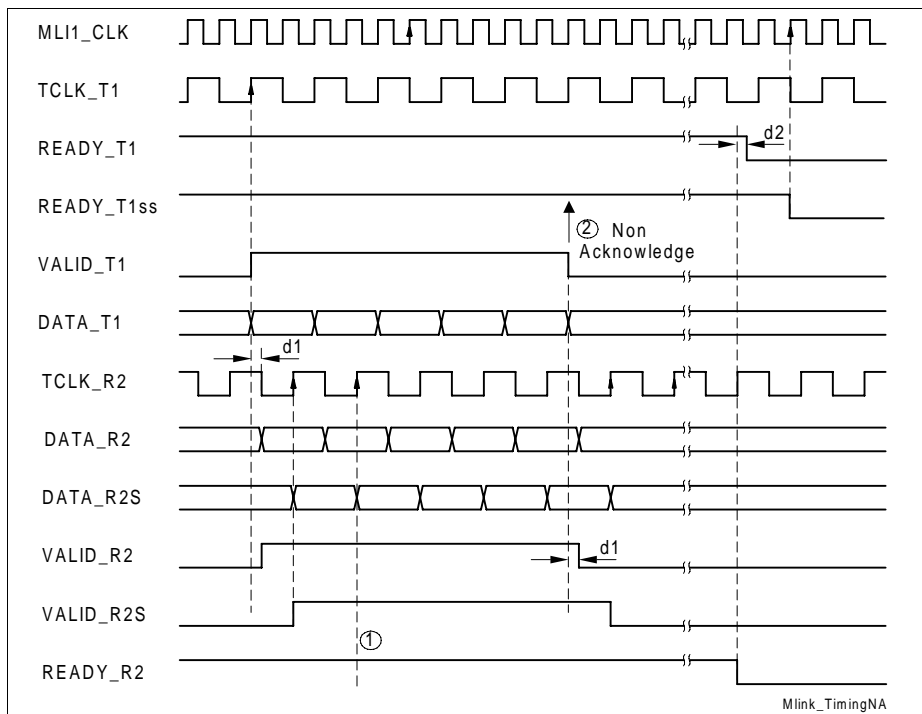


Figure 7-40 Detailed Handshake in a Transfer with Non Acknowledge Error

In figure above number 1 represents the moment in which the MLI receiver should set its READY signal. Number 2 represents the instant in which the MLI transmitter checks the READY signal status.

A non acknowledge situation may lead into a time out one if the MLI receiver does not raise again the READY signal before the counter of non acknowledge errors overflow.

7.2.11 Data Flow Description

7.2.11.1 Copy Base Address

The copy base address frame is used to transmit the two parameters of a remote window for pipe x, the 28 most significant base address bits and the 4-bit coded buffer size, from the local microcontroller to the remote microcontroller.

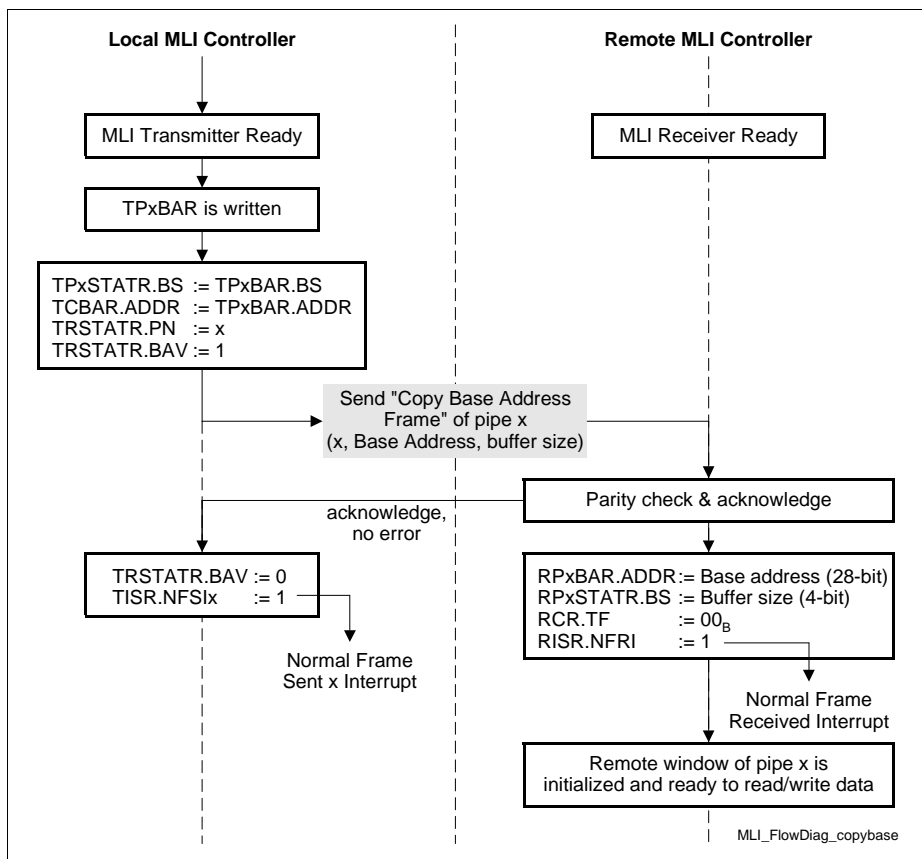


Figure 7-41 Copy Base Address Frame Flow

The transmission of a copy base address frame is initiated by writing the two parameters for a pipe remote window, the 28 most significant base address bits and the 4-bit coded remote window (buffer) size, into register TPx.BAR.

Micro Link Serial Bus Interface (MLI)

7.2.11.2 Command Frame

The transmission of a command frame is initiated by writing one of the four pipe x related command code bit fields in register TCMDR. Depending on the pipe x related command code that is transmitted, different actions are triggered in the remote microcontroller.

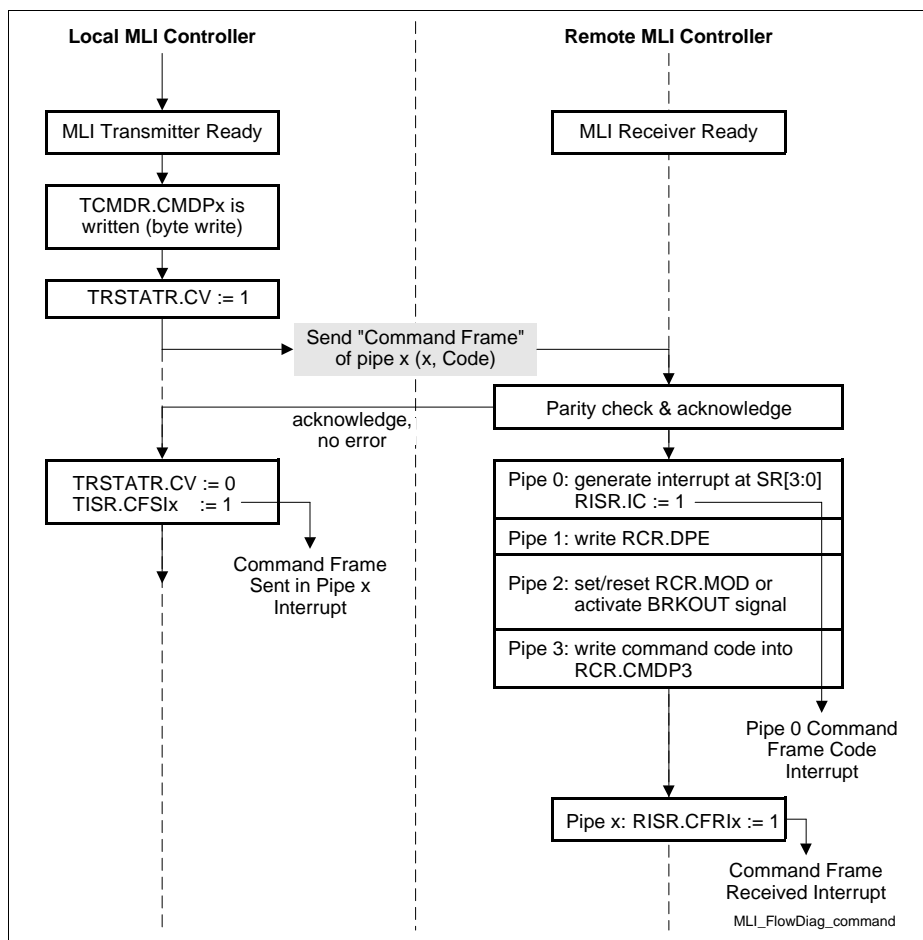


Figure 7-42 Command Frame Flow

Micro Link Serial Bus Interface (MLI)

7.2.11.3 Write Frame

The transmission of a write frame is initiated in the local microcontroller by a write access of a bus master (e.g. the CPU) to one of the eight transfer windows.

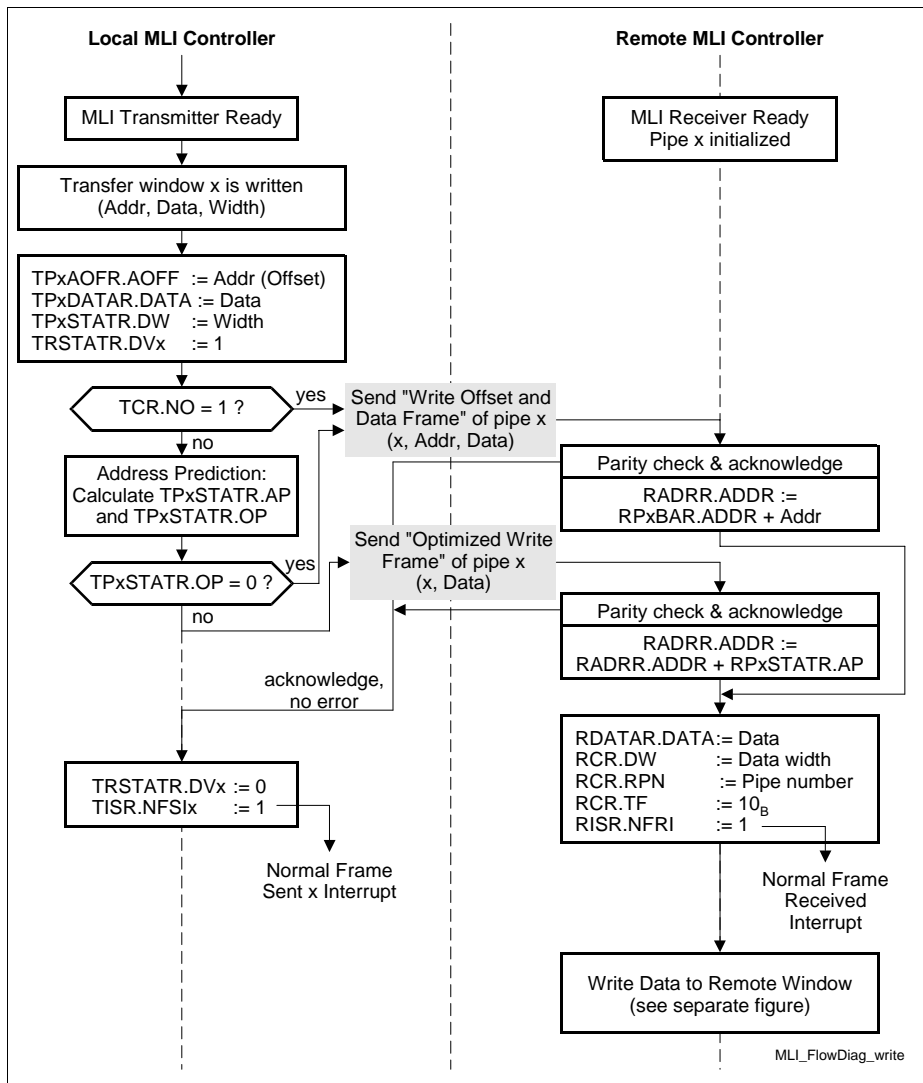


Figure 7-43 Write Frame Flow

7.2.11.4 Read Frame

The transmission of a read frame (see [Figure 7-44](#)) is initiated in the local microcontroller by a read access of a bus master (e.g. the CPU) to one of the eight transfer windows.

Micro Link Serial Bus Interface (MLI)

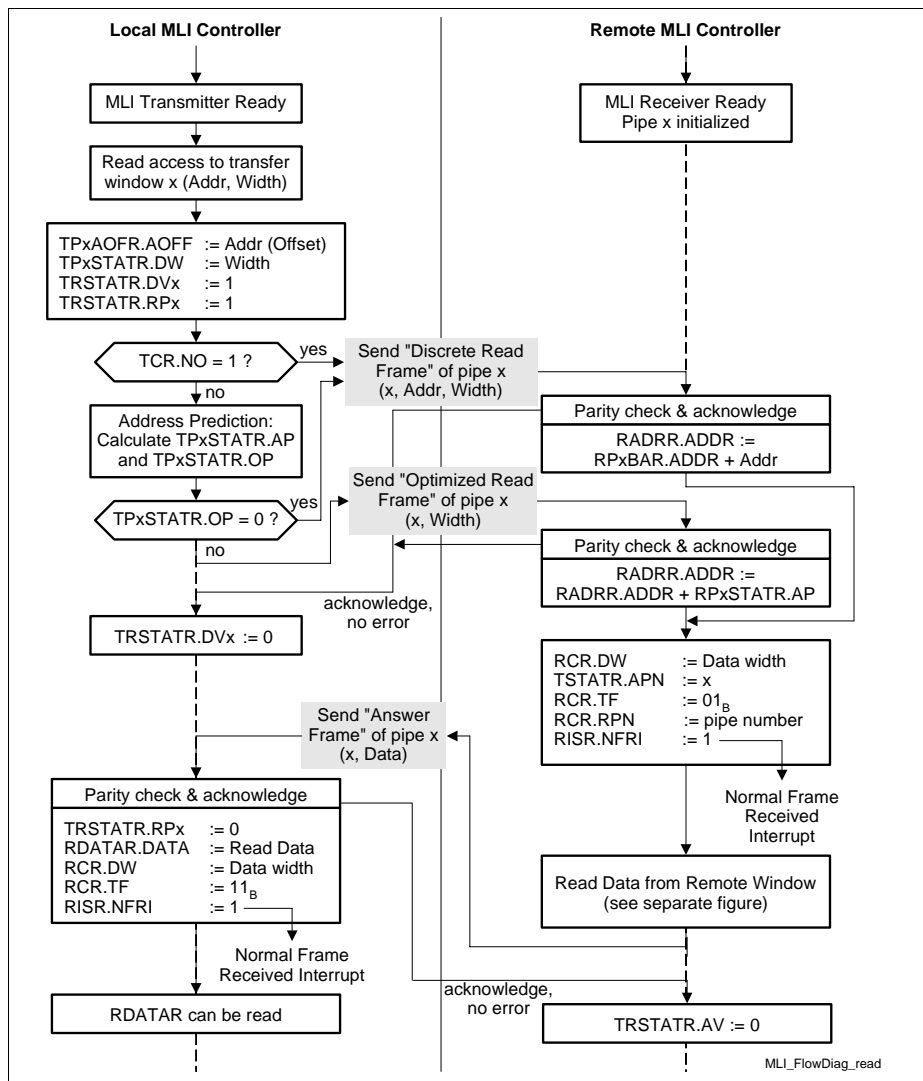


Figure 7-44 Read Frame Flow

7.2.11.5 Access to Remote Window

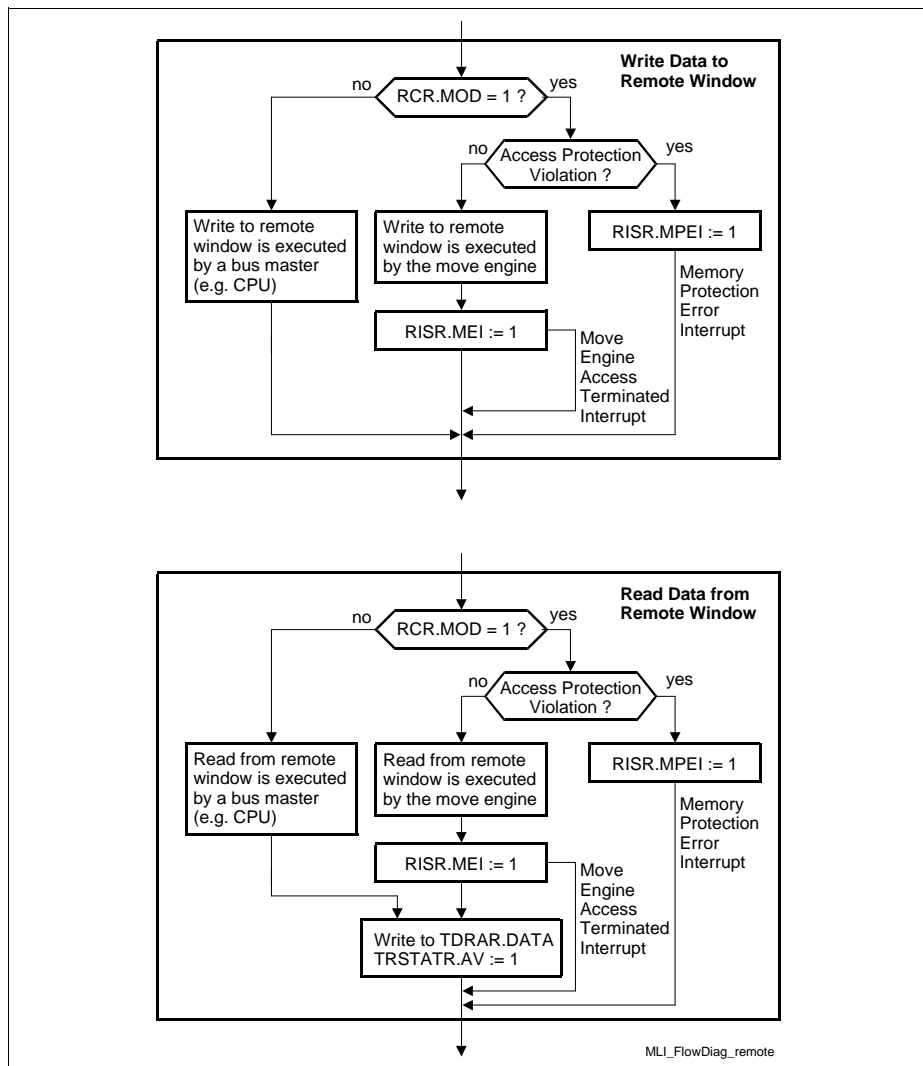


Figure 7-45 Access to Remote Window

Micro Link Serial Bus Interface (MLI)

7.2.12 Baud Rate Generation

This chapter describes the baud rate generation for the MLI module.

The MLI transmitter baud rate is given by $f_{\text{MLI}}/2$, with f_{MLI} being programmable by the fractional divider FDIV.

The receiver baud rate is determined by the connected transmitter. The receiver frequency should not exceed $f_{\text{SYS}}/2$.

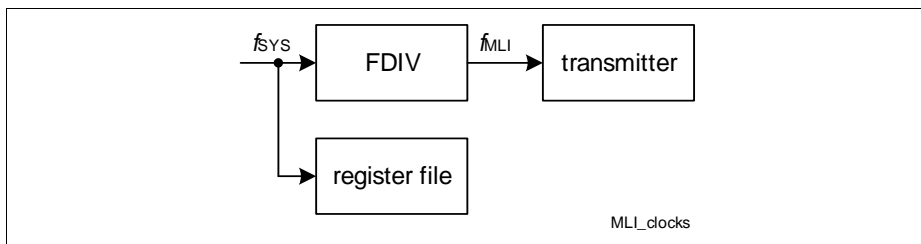


Figure 7-46 MLI Baud Rate Generation

7.2.12.1 Fractional Divider

Functional Description

The fractional divider allows to generate a clock enable depending from an input clock f_{HW_CLK} using a programmable divider. The fractional divider divides the input clock f_{HW_CLK} either by the factor $1/n$ or by a fraction of $n/1024$ for any value of n from 0 to 1023.

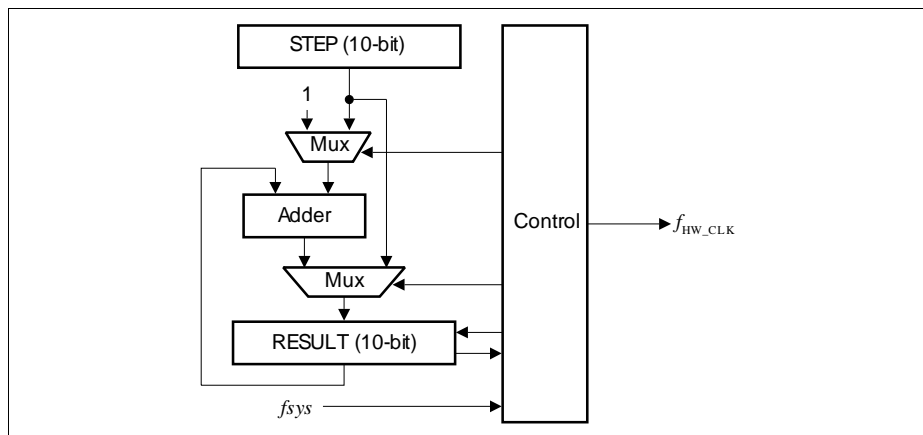


Figure 7-47 Fractional Divider Block Diagram

The fractional divider has two operating modes:

- Normal Divider Mode
- Fractional Divider Mode

Micro Link Serial Bus Interface (MLI)

Normal Divider Mode

In Normal Divider Mode (**FDR.DM** = 01_B) the fractional divider is a simple linear divider. **FDR.RESULT** represents the counter value and **FDR.STEP** defines the reload value.

The frequency in normal divider mode is defined according the following formula:

$$f_{\text{mod}} = f_{\text{HW_CLK}} \times \frac{1}{n} \quad \text{with } n = 1024 - \text{STEP} \quad (7.1)$$

In order to get $f_{\text{mod}} = f_{\text{HW_CLK}}$ STEP must be programmed with 3FF_H. **Figure 7-48** shows the operation of the normal divider mode with a reload value of **FDR.STEP** = 3FD_H.

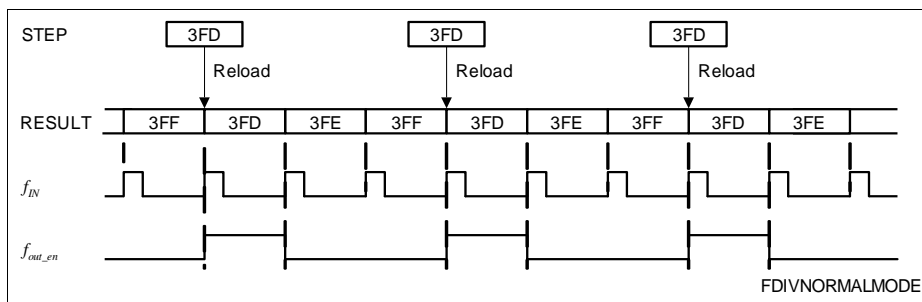


Figure 7-48 Normal Divider Mode Example

Micro Link Serial Bus Interface (MLI)

Fractional Divider Mode

When the Fractional Divider Mode is selected (**FDR.DM** = 10_B), the clock is derived from the input clock $f_{\text{HW_CLK}}$ by multiplication of fraction $n/1024$ ($n = 0 \dots 1023$).

The Fractional Divider Mode frequency is defined according the following formula:

$$f_{\text{mod}} = f_{\text{HW_CLK}} \times \frac{n}{1024} \quad \text{with } n = 0 - 1023 \quad (7.2)$$

Figure 7-49 shows the operation of the Fractional Divider Mode with a reload value of **FDR.STEP** = 234_H ($f_{\text{mod}} = f_{\text{HW_CLK}} * 564/1024 = 0.55$).

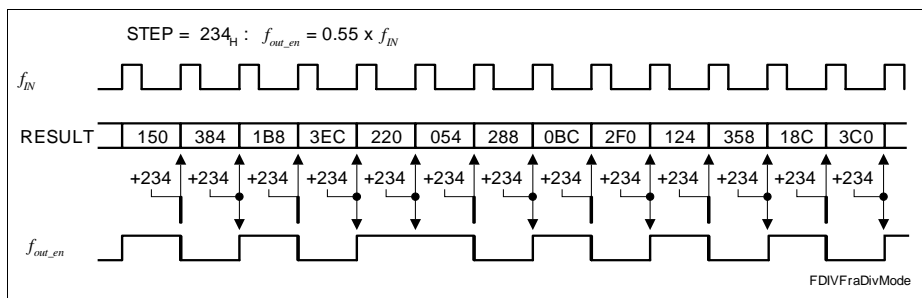


Figure 7-49 Fractional Divider Mode Example

Micro Link Serial Bus Interface (MLI)

7.3 MLI Kernel Registers

Figure 7-50 and Table 7-27 show all registers associated with the MLI Kernel.

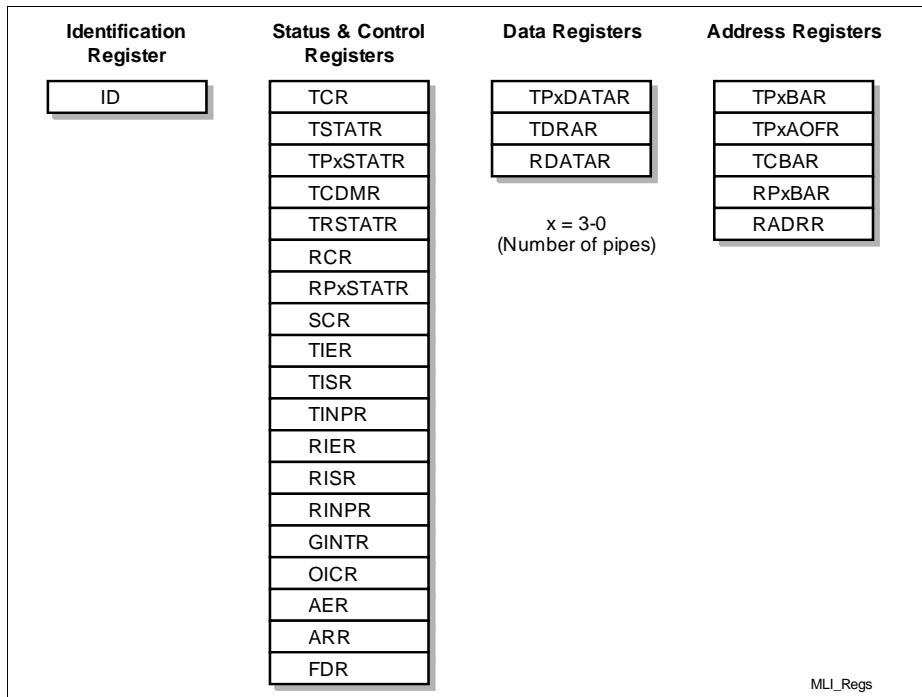


Figure 7-50 MLI Kernel Registers

Note: The letter “x” indicates the number of pipe (pipes 0, 1, 2 and 3).

Note: All bits marked ‘w’ return 0 when read.

Table 7-26 MLI Kernel Register Address Space

Module	Base Address	End Address	Note
MLI	0000 0200 _H	0000 02FF _H	256 byte

Table 7-27 MLI Kernel Registers

Register Short Name	Register Long Name	Offset Address	Description see
ID	Module Identification Register	0008 _H	Page 7-67
FDR	Fractional Divider Register	000C _H	Page 7-110
TCR	Transmitter Control Register	0010 _H	Page 7-68
TSTATR	Transmitter Status Register	0014 _H	Page 7-71
TP0STATR	Transmitter Pipe 0 Status Register	0018 _H	Page 7-73
TP1STATR	Transmitter Pipe 1 Status Register	001C _H	Page 7-73
TP2STATR	Transmitter Pipe 2 Status Register	0020 _H	Page 7-73
TP3STATR	Transmitter Pipe 3 Status Register	0024 _H	Page 7-73
TCMDR	Transmitter Command Register	0028 _H	Page 7-75
TRSTATR	Transmitter Registers Status Register	002C _H	Page 7-77
TP0AOFR	Transmitter Pipe 0 Address Offset Register	0030 _H	Page 7-79
TP1AOFR	Transmitter Pipe 1 Address Offset Register	0034 _H	Page 7-79
TP2AOFR	Transmitter Pipe 2 Address Offset Register	0038 _H	Page 7-79
TP3AOFR	Transmitter Pipe 3 Address Offset Register	003C _H	Page 7-79
TP0DATAR	Transmitter Pipe 0 Data Register	0040 _H	Page 7-80
TP1DATAR	Transmitter Pipe 1 Data Register	0044 _H	Page 7-80
TP2DATAR	Transmitter Pipe 2 Data Register	0048 _H	Page 7-80
TP3DATAR	Transmitter Pipe 3 Data Register	004C _H	Page 7-80
TDRAR	Transmitter Data Read Answer Register	0050 _H	Page 7-81
TP0BAR	Transmitter Pipe 0 Base Address Register	0054 _H	Page 7-82
TP1BAR	Transmitter Pipe 1 Base Address Register	0058 _H	Page 7-82
TP2BAR	Transmitter Pipe 2 Base Address Register	005C _H	Page 7-82
TP3BAR	Transmitter Pipe 3 Base Address Register	0060 _H	Page 7-82
TCBAR	Transmitter Copy Base Address Register	0064 _H	Page 7-83
RCR	Receiver Control Register	0068 _H	Page 7-84
RP0BAR	Receiver Pipe 0 Base Address Register	006C _H	Page 7-87
RP1BAR	Receiver Pipe 1 Base Address Register	0070 _H	Page 7-87
RP2BAR	Receiver Pipe 2 Base Address Register	0074 _H	Page 7-87
RP3BAR	Receiver Pipe 3 Base Address Register	0078 _H	Page 7-87

Micro Link Serial Bus Interface (MLI)

Table 7-27 MLI Kernel Registers (cont'd)

Register Short Name	Register Long Name	Offset Address	Description see
RP0STATR	Receiver Pipe 0 Status Register	007C _H	Page 7-88
RP1STATR	Receiver Pipe 1 Status Register	0080 _H	Page 7-88
RP2STATR	Receiver Pipe 2 Status Register	0084 _H	Page 7-88
RP3STATR	Receiver Pipe 3 Status Register	0088 _H	Page 7-88
RADDR	Receiver Address Register	008C _H	Page 7-89
RDATAR	Receiver Data Register	0090 _H	Page 7-90
SCR	Set Clear Register	0094 _H	Page 7-91
TIER	Transmitter Interrupt Enable Register	0098 _H	Page 7-93
TISR	Transmitter Interrupt Status Register	009C _H	Page 7-95
TINPR	Transmitter Interrupt Node Pointer Register	00A0 _H	Page 7-96
RIER	Receiver Interrupt Enable Register	00A4 _H	Page 7-98
RISR	Receiver Interrupt Status Register	00A8 _H	Page 7-100
RINPR	Receiver Interrupt Node Pointer Register	00AC _H	Page 7-101
GINTR	Global Interrupt Set Register	00B0 _H	Page 7-103
OICR	Output Input Control Register	00B4 _H	Page 7-104
-	-	00B8 _H	-
-	-	00BC _H	-

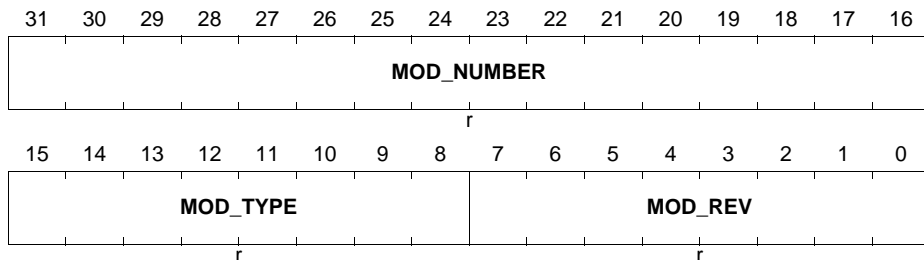
7.3.1 Module Identification Register

ID

Module Identification Register

(08_H)

Reset Value: 0025 C006_H



Field	Bits	Type	Description
MOD_REV	[7:0]	r	Module Revision Number MOD_REV defines the module revision number. The value of a module revision starts with 01 _H (first revision). Current Version: 06 _H
MOD_TYPE	[15:8]	r	Module Type The value of this bit field is C0 _H . It defines the module as a 32-bit module.
MOD_NUMBER	[31:16]	r	Module Number Value This bit field defines a module identification number. For the MLI module the module identification number is 25 _H .

Micro Link Serial Bus Interface (MLI)

7.3.2 MLI Transmitter Registers

7.3.2.1 Transmitter Control Register

TCR

Transmitter Control Register

(10_H)

Reset Value: 0000 0110_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP	NO	MDP				MNAE		MPE				0	RTY	DNT	MOD
rw	rw	rw				rwh		rwh				r	rw	rw	rw

Field	Bits	Type	Description
MOD	0	rw	Mode of Operation This bit establishes the operation mode of the MLI transmitter. Its encoding is as follows: 0 MLI transmitter off 1 MLI transmitter on
DNT	1	rw	Data in Not Transmission This bit will determine the level of the data line when no transmission is in progress. If set to one, the data line when the transmission is finished will have the value 1. If set to zero the DATA line level will be 0.
RTY	2	rw	Retry This bit enables the retry mechanism for the transfer windows. 0 The retry mechanism is disabled. Any access while the transmitter is busy is discarded without additional action. 1 The retry mechanism is enabled. Any access while the transmitter is busy is acknowledged with a retry. In this case, the requesting bus master sends the requested access again until the request is accepted.

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
MPE	[7:4]	rwh	Maximum Parity Errors This bit field indicates the value of parity errors for the parity interrupt generation. It is set to its maximum value by software and its number will be decreased by the MLI, each time it detects a parity error. Its encoding is as follows: 0000 _B Generate parity error interrupt 0001 _B 1 parity error for interrupt generation 0010 _B 2 parity errors for interrupt generation ... 1111 _B 15 parity errors for interrupt generation
MNAE	[9:8]	rwh	Maximum Non Acknowledge Errors This bit field indicates the number of acknowledge errors for the time out interrupt generation. It is set to its maximum value by software and its number will be decreased by the MLI, each time it detects a non acknowledge error. Its encoding is as follows: 00 _B Generate non acknowledge error interrupt 01 _B One error for the interrupt generation 10 _B Two errors for the interrupt generation 11 _B Three errors for the interrupt generation If the non acknowledge error does not appear again before the counter reaches zero, the MLI resets again this field to 11 _B .
MDP	[13:10]	rw	Maximum Delay for Parity Error This bit field is written by software and it defines the number of clock periods (from the clock used in the transmission) above which if the READY signal remains low it will be considered parity error condition. These bits will be interpreted as follows: 0000 _B Zero clock periods 0001 _B One clock period ... 1110 _B Fourteen clock periods 1111 _B Fifteen clock periods

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
NO	14	rw	No Optimized Method This bit field indicates if the optimized method for address prediction is enabled or not. Its encoding is as follows: 0 Optimized method enabled 1 Optimized method not enabled
TP	15	rw	Type of Parity This bit will determine the type of parity used in the transmission. This value will be the one initially used for the toggle bit that produces the parity bit. Assuming a correct transmission, when set to one it will force the MLI receiver to produce a parity error condition.
0	3, [31:16]	r	Reserved ; returns 0 if read; should be written with 0.

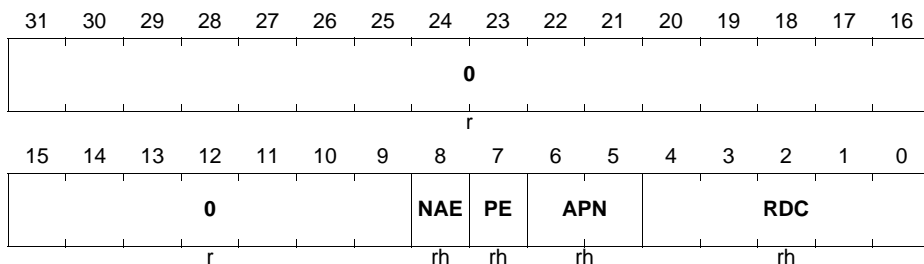
7.3.2.2 Transmitter Status Register

TSTATR

Transmitter Status Register

(14_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
RDC	[4:0]	rh	Ready Delay Counter This counter is reset to zero when the VALID signal goes low-level after a transmission, and it will count TCLK periods until the MLI transmitter detects that the READY signal is high level again, or when the counter reaches its maximum value.
APN	[6:5]	rh	Answer Pipe Number This bit field is written by the MLI receiver whenever it gets a read frame. This value will be coincident with the pipe number of the received read frame and it will be used to send the answer frame. Its encoding is as follows: 00 _B Send the answer frame through pipe 0 01 _B Send the answer frame through pipe 1 10 _B Send the answer frame through pipe 2 10 _B Send the answer frame through pipe 3
PE	7	rh	Parity Error Flag Set to one when the MLI transmitter detects a parity error in the transmission. It is reset again when the MLI makes a transfer without parity error or when set the SCR.CTPE bit.

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
NAE	8	rh	Non Acknowledge Error Flag Set to one when the MLI transmitter detects a non acknowledge error in the transmission. It is reset again when the MLI makes a transfer without error or when set the SCR.CNAE bit.
0	[31:9]	r	Reserved ; returns 0 if read; should be written with 0.

7.3.2.3 Transmitter Pipe Status Registers

TP0STATR

Transmitter Pipe 0 Status Register (18_H) **Reset Value: 0000 0000_H**

TP1STATR

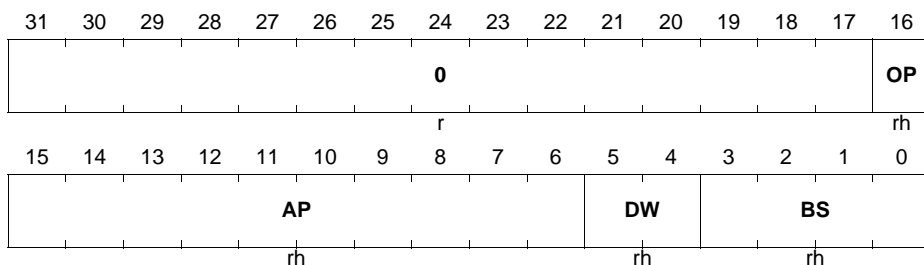
Transmitter Pipe 1 Status Register (1C_H) **Reset Value: 0000 0000_H**

TP2STATR

Transmitter Pipe 2 Status Register (20_H) **Reset Value: 0000 0000_H**

TP3STATR

Transmitter Pipe 3 Status Register (24_H) **Reset Value: 0000 0000_H**



Field	Bits	Type	Description
BS	[3:0]	rh	Buffer Size It gives the size of the transfer window in the second microcontroller. The offset width will be coincident with this buffer size field. This field is updated by the MLI transmitter when the correspondent TPxBAR register is written. 0000 _B One bit offset 0001 _B Two bit offset ... 1111 _B Sixteen bits offset

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
DW	[5:4]	rh	Data Width This bit field defines the width of the data written in the TPxDATAR register. It is written by the MLI transmitter each time a new data is received in the TPxDATAR register. 00 _B Data width of 8 bits selected 01 _B Data width of 16 bits selected 10 _B Data width of 32 bits selected 11 _B Reserved
AP	[15:6]	rh	Address Prediction Factor It is written by the MLI transmitter. It is used to keep track of the address prediction method. It represents a number of ten bits with sign in two's complement.
OP	16	rh	Use Optimized Frame This bit field is written by the MLI transmitter each time it performs the necessary operations to know if the address offset follows the address prediction scheme. Its meaning is as follows: 0 Do not use optimized mode to send the frame 1 Use optimized mode to send the frame It is used by the MLI to know if the optimized method should or not be used when a frame is selected to be sent.
0	[31:17]	r	Reserved ; returns 0 if read; should be written with 0.

Micro Link Serial Bus Interface (MLI)

7.3.2.4 Transmitter Command Register

This register must be written only bite-wisely. Write accesses with a larger data width than a byte are forbidden. A new command can only be started if the one before is either finished or has been aborted. There must not be more than one command request pending at the same time.

TCMDR

Transmitter Command Register

(28_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				CMDP3				0				CMDP2			
r				rw				r				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				CMDP1				0				CMDP0			
r				rw				r				rw			

Field	Bits	Type	Description
CMDP0	[3:0]	rw	Command in pipe 0 This bit field is written by software and it defines the command to be sent through pipe 0. These bits will be interpreted as follows: 0000 _B Generate interrupt 0, if enabled by RIER .ICE 0001 _B Generate interrupt 1, if enabled by RIER .ICE 0010 _B Generate interrupt 2, if enabled by RIER .ICE 0011 _B Generate interrupt 3, if enabled by RIER .ICE others No effect
CMDP1	[11:8]	rw	Command in pipe 1 This bit field is written by software and it defines the command to be sent through pipe 1. These bits will be interpreted as follows: 0000 _B Make RCR .DPE = 0000 _B 0001 _B Make RCR .DPE = 0001 _B ... 1111 _B Make RCR .DPE = 1111 _B
CMDP2	[19:16]	rw	Command in pipe 2 This bit field is written by software and it defines the command to be sent through pipe 2, see Table 7-22 “Command Frame Encoding” on Page 7-38 .

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
CMDP3	[27:24]	rw	Command in pipe 3 This bit field is written by software and it defines the command to be sent through pipe 3. The commands will be software interpreted.
0	[7:4], [15:12], [23:20], [31:28]	r	Reserved ; returns 0 if read; should be written with 0.

7.3.2.5 Transmitter Registers Status Register

TRSTATR

Transmitter Registers Status Register(2C_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0						PN	RP3	RP2	RP1	RP0	DV3	DV2	DV1	DV0	
r						rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						BAV	AV	CV3	CV2	CV1	CV0	CIV3	CIV2	CIV1	CIV0
r						rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
CIVx (x = 0-3)	x	rh	Command Interrupt Valid Set to one by the MLI transmitter whenever it detects a rising edge in the correspondent trigger_commandx line. It is reset again when the correspondent command frame to the line is correctly sent through pipe 0, or when set SCR.CCIVx .
CVx (x = 0-3)	x+4	rh	Command Valid Set to one by the MLI transmitter when the TCMDR.CMDPx bit field is written, or when set SCR.SCVx . It is reset again when the command frame is correctly sent, or when set SCR.CCVx .
AV	8	rh	Answer Valid Set to one by the MLI transmitter when the TDRAR register is written. It is reset again when the answer frame is correctly sent, or when set SCR.CAV .
BAV	9	rh	Base Address Valid Set to one by the MLI transmitter when the TCBAR register is written. It is reset again when the copy base address frame is correctly sent, or when set SCR.CBAV .

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
DVx (x = 0-3)	x+16	rh	Data Valid Set to one by the MLI transmitter when the TPxDATAR and/or the TPxOFR registers are written. It is reset again when the read or write frame is correctly sent, or when set SCR.CDVx .
RPx (x = 0-3)	x+20	rh	Read Pending Set to one by the MLI transmitter when the TPxOFR register is written for a read operation. It is reset again when MLI receiver gets an answer frame for pipe x or when SCR.CRPx is set.
PN	[25:24]	rh	Pipe Number This bit field will indicate to which pipe number corresponds the base address that has been stored in the TCBAR register, and it is written by the MLI transmitter each time it writes TCBAR with the data from the correspondent TPxBAR register. Its encoding is as follows: 00 _B Pipe number 0 01 _B Pipe number 1 01 _B Pipe number 2 11 _B Pipe number 3
0	[15:10], [31:26]	r	Reserved ; returns 0 if read; should be written with 0.

7.3.2.7 Transmitter Pipe Data Registers

TP0DATAR

Transmitter Pipe 0 Data Register (40_H) Reset Value: 0000 0000_H

TP1DATAR

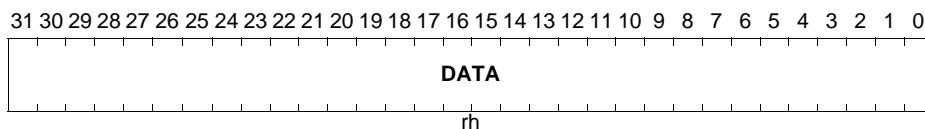
Transmitter Pipe 1 Data Register (44_H) Reset Value: 0000 0000_H

TP2DATAR

Transmitter Pipe 2 Data Register (48_H) Reset Value: 0000 0000_H

TP3DATAR

Transmitter Pipe 3 Data Register (4C_H) Reset Value: 0000 0000_H



Field	Bits	Type	Description
DATA	[31:0]	rh	Data Contains the data that will be sent to the MLI receiver in the other microcontroller through the correspondent pipe. It will be the data to be written and to send it the MLI will use a write frame (optimized or not).

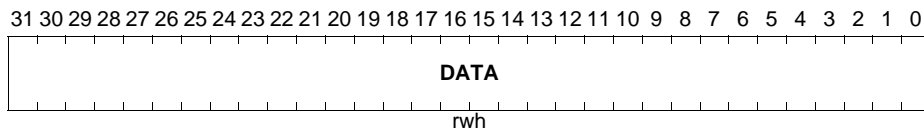
Micro Link Serial Bus Interface (MLI)

7.3.2.8 Transmitter Data Read Answer Register

TDRAR

Transmitter Data Read Answer Register (50_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
DATA	[31:0]	rwh	Data Contains the data that proceeds from a read operation and it will be sent to the MLI receiver in the other microcontroller. It will be the data that is sent in answer frame.

Micro Link Serial Bus Interface (MLI)

7.3.2.9 Transmitter Pipe Base Address Registers

When almost the least significant byte of any of the next registers is updated, the 28 MSB's are directly copied in the register TCBAR, and the BS bit field is copied in the correspondent TPxSTATR register.

TP0BAR

Transmitter Pipe 0 Base Address Register(54_H)

Reset Value: 0000 0000_H

TP1BAR

Transmitter Pipe 1 Base Address Register(58_H)

Reset Value: 0000 0000_H

TP2BAR

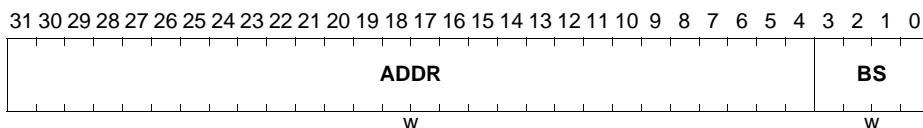
Transmitter Pipe 2 Base Address Register(5C_H)

Reset Value: 0000 0000_H

TP3BAR

Transmitter Pipe3 Base Address Register(60_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
BS	[3:0]	w	Buffer Size It gives the used size of the remote window in the remote microcontroller. The offset width will be coincident with this buffer size field. 0000 _B One bit offset 0001 _B Two bits offset 0010 _B Three bits offset ... 1110 _B Fourteen bits offset 1111 _B Sixteen bits offset
ADDR	[31:4]	w	Address This bit field contains the base address 28 MSB's of the correspondent pipe. It is written each time the microcontroller wanted to initialize a pipe in the other microcontroller.

Micro Link Serial Bus Interface (MLI)

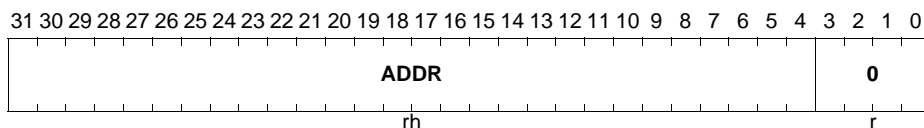
7.3.2.10 Transmitter Copy Base Address Register

This register is updated with the 28 MSB's contained in the latest accessed TPxBAR register when one of those is accessed. The trigger condition to make the copy is that the least significative byte of any of the TPxBAR registers had been written.

TCBAR

Transmitter Copy Base Address Register(64_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
ADDR	[31:4]	rh	Address This bit field contains the base address 28 MSB's of any of the four pipes.
0	[3:0]	r	Reserved ; returns 0 if read; should be written with 0.

*Note: The **TRSTATR.BAVx** flag indicates if this register contains a base address that has not still sent. In fact that flag is reset again when the copy base address frame has been correctly sent.*

Micro Link Serial Bus Interface (MLI)

7.3.3 MLI Receiver Registers

7.3.3.1 Receiver Control Register

RCR

Receiver Control Register

(68_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							RCV RST	0			BEN	MPE			
r							rw	r			rw	rwh			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPN		PE	TF		DW		MOD	CMDP3				DPE			
rh		rh	rh		rh		rh	rh				rh			

Field	Bits	Type	Description
DPE	[3:0]	rh	Delay for Parity Error This bit field is written by the MLI when it receives the proper command to program it. It defines the number of clock periods (from the clock used in the transmission) that as minimum the MLI receiver has to wait to raise again the READY signal when it has detected a parity error. 0000 _B Wait zero clock periods 0001 _B Wait one clock period 0010 _B Wait two clock periods ... 1110 _B Wait fourteen clock periods 1111 _B Wait fifteen clock periods
CMDP3	[7:4]	rh	Command From Pipe 3 Whenever the MLI receiver gets a command frame trough the pipe 3, it stores its value in this bit field. The command will be interpreted by software.

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
MOD	8	rh	<p>Mode of Operation</p> <p>This bit field defines the mode of operation of the MLI receiver. The MLI receiver may write this bit field when it receives the proper command from other microcontroller.</p> <p>This bit is also set and reset when its correspondent bits of the clear and set register are set, SCR.CMOD and SCR.SMOD.</p> <p>0 The automatic read/write handling is disabled. The request is stored in the registers and can be served by the CPU (listening mode).</p> <p>1 The automatic read/write handling is enabled.</p>
DW	[10:9]	rh	<p>Data Width</p> <p>This bit field is updated by the MLI receiver whenever it writes new data in the RDATAR register. Bit field DW is used by the MLI receiver when it delivered the data or by the software when ever it had to fetch data from the MLI receiver RDATAR register. It is updated by read frames, write frames or answer frames.</p> <p>00_B Data width of 8 bits selected 01_B Data width of 16 bits selected 10_B Data width of 32 bits selected 11_B Reserved</p>
TF	[12:11]	rh	<p>Type of Frame</p> <p>Set by the MLI receiver when it writes the RDATAR, RADRR and RPxBAR registers. Its value depends on the kind of frame in which the data was received. This bit field will be used by the software in order to know where it must take the data from and how to deliver it. It is updated by copy base address frames, read frames, write frames or answer frames.</p> <p>00_B Copy base address frame 01_B Read frame, optimized or not 10_B Write frame, optimized or not 11_B Answer frame</p>
PE	13	rh	<p>Parity Error</p> <p>Set to one when the MLI receiver detects a parity error in the transmission. It is reset again when the MLI receives a transfer without parity error, or when set SCR.CRPE.</p>

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
RPN	[15:14]	rh	Received Pipe Number This bit field contains the pipe number that was indicated by the pipe number bit field of the latest received frame. It is updated by copy base address frames, read frames, write frames or answer frames.
MPE	[19:16]	rwh	Maximum Parity Errors This bit field indicates after how many parity errors the parity error interrupt will be generated. It is set to a desired value by software and it is decremented automatically by the MLI, each time it detects a parity error. If 0, each parity error will generate an interrupt and MPE stays 0. 0000 _B Each parity error will generate the interrupt. 0001 _B After 2 parity errors the interrupt is generated. 0010 _B After 3 parity errors the interrupt is generated. ... 1111 _B After 16 parity errors the interrupt is generated.
BEN	20	rw	Break Out Enable If set to one the MLI receiver will produce a pulse in its BREAKOUT line when received the corresponding command frame.
RCVRST	24	rw	Receiver Reset This bit forces the receiver to reset in order to be able to change OICR settings without influencing the receiver register. During chip reset, this bit is 1 and in the second clock cycle after the chip reset, the specified reset value is applied. This ensures that no spikes/pulses are received during chip reset (OICR might change). 0 The receiver is not held in reset (operating mode). 1 The receiver is held in reset.
0	[23:21], [31:25]	r	Reserved ; returns 0 if read; should be written with 0.

7.3.3.2 Receiver Pipe Base Address Registers

RP0BAR

Receiver Pipe 0 Base Address Register (6C_H)

Reset Value: 0000 0000_H

RP1BAR

Receiver Pipe 1 Base Address Register (70_H)

Reset Value: 0000 0000_H

RP2BAR

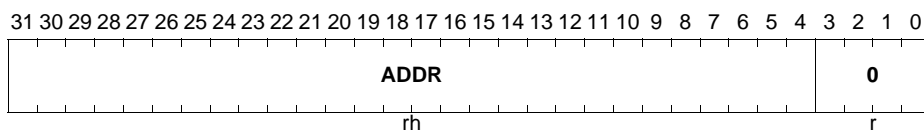
Receiver Pipe 2 Base Address Register (74_H)

Reset Value: 0000 0000_H

RP3BAR

Receiver Pipe 3 Base Address Register (78_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
ADDR	[31:4]	rh	Address ADDR indicates the pipe x remote window base address. When a pipe x copy base address frame is received, ADDR[31:4] becomes loaded with the transmitted 28-bit address and bits [3:0] are loaded with 0000 _B . For address calculation, ADDR[RPxSTATR.BS-1:0] is used for calculation of the actual read/write address that is stored in register RADDR. ADDR[31:RPxSTATR.BS] remains frozen for this address calculation.
0	[3:0]	r	Reserved ; returns 0 if read; should be written with 0.

Micro Link Serial Bus Interface (MLI)

7.3.3.3 Receiver Pipe Status Registers

RP0STATR

Receiver Pipe 0 Status Register (7C_H) Reset Value: 0000 0000_H

RP1STATR

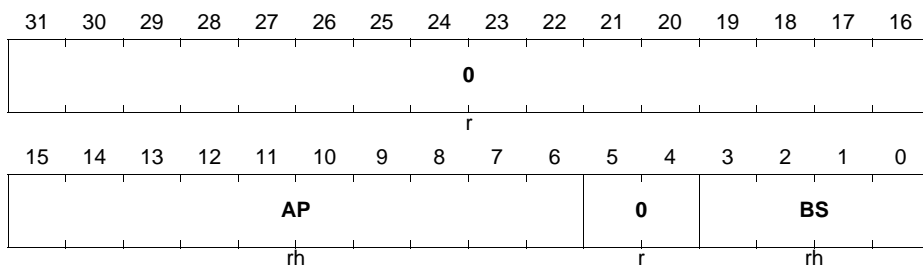
Receiver Pipe 1 Status Register (80_H) Reset Value: 0000 0000_H

RP2STATR

Receiver Pipe 2 Status Register (84_H) Reset Value: 0000 0000_H

RP3STATR

Receiver Pipe 3 Status Register (88_H) Reset Value: 0000 0000_H



Field	Bits	Type	Description
BS	[3:0]	rh	Buffer Size It is written by the MLI receiver each time it receives a copy base address frame. The offset width of the offset field received in the frames will be coincident with this buffer size field. 0000 _B One bit offset 0001 _B Two bits offset 0010 _B Three bits offset ... 1110 _B Fourteen bits offset 1111 _B Sixteen bits offset
AP	[15:6]	rh	Address Prediction Factor It is written by the MLI receiver. It is used to keep track of the address prediction method. It represents a number of ten bits with sign in two's complement.
0	[5:4], [31:16]	r	Reserved ; returns 0 if read; should be written with 0.

Micro Link Serial Bus Interface (MLI)

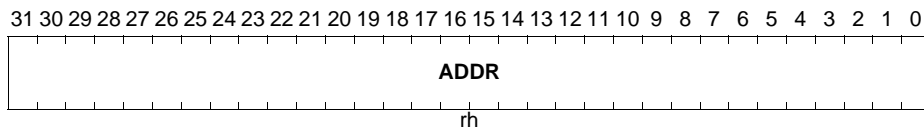
7.3.3.4 Receiver Address Register

RADRR

Receiver Address Register

(8C_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
ADDR	[31:0]	rh	Address It contains the destination absolute address in which the data will be written or from which the data will be read. Its value is totally synchronous to the MLI receiver internal clock.

Micro Link Serial Bus Interface (MLI)

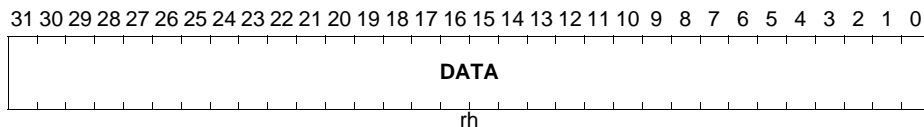
7.3.4 Receiver Data Register

RDATAR

Receiver Data Register

(90_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
DATA	[31:0]	rh	Data It contains the data that will be written in the microcontroller. It can be data of a write operation or the result from a read operation. Its value is totally synchronous to the MLI receiver internal clock.

7.3.5 MLI Kernel Common Registers

7.3.5.1 Set Clear Register

SCR

Set Clear Register

(94_H)

Reset Value: 0000 0010_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCIV 3	CCIV 2	CCIV 1	CCIV 0	CNA E	CTP E	CRP E	CAV	0						CBA V	CMO D
w	w	w	w	w	w	w	w	r						w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCV 3	CCV 2	CCV 1	CCV 0	CDV 3	CDV 2	CDV 1	CDV 0	0			SMO D	SCV 3	SCV 2	SCV 1	SCV 0
w	w	w	w	w	w	w	w	r			w	w	w	w	w

Field	Bits	Type	Description
SCVx (x = 0-3)	x	w	Set Command Valid 0 No effect. 1 Bit TRSTATR .CVx is set independently on the value written to CCVx.
SMOD	4	w	Set MOD Flag 0 No effect. 1 Bit RCR .MOD is set.
CDVx (x = 0-3)	x+8	w	Clear Data Valid 0 Flag 0 No effect. 1 Bits TRSTATR .DVx and TRSTATR .RPx are cleared.
CCVx (x = 0-3)	x+12	w	Clear Command Valid 0 Flag 0 No effect. 1 Bit TRSTATR .CVx is cleared.
CMOD	16	w	Clear MOD Flag 0 No effect. 1 Bit RCR .MOD is cleared.
CBAV	17	w	Clear BAV Flag 0 No effect. 1 Bit TRSTATR .BAV is cleared.

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
CAV	24	w	Clear AV Flag 0 No effect. 1 Bit TRSTATR .AV is cleared.
CRPE	25	w	Clear Receiver PE Flag 0 No effect. 1 Bit RCR .PE is cleared.
CTPE	26	w	Clear Transmitter PE Flag 0 No effect. 1 Bit TSTATR .PE is cleared.
CNAE	27	w	Clear NAE Flag 0 No effect. 1 Bit TSTATR .NAE is cleared.
CCIVx (x = 0-3)	x+28	w	Clear Command Interrupt Valid x Flag 0 No effect. 1 Bit TRSTATR .CIVx is cleared.
0	[7:5], [23:18]	r	Reserved ; returns 0 if read; should be written with 0.

*Note: Reading register **SCR** always returns zeros at all bit locations.*

7.3.6 MLI Interrupt Registers

7.3.6.1 Transmitter Interrupt Enable Register

TIER

Transmitter Interrupt Enable Register (98_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0						TE IR	PE IR	CFS IR3	CFS IR2	CFS IR1	CFS IR0	NFS IR3	NFS IR2	NFS IR1	NFS IR0
r						w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						TE IE	PE IE	CFS IE3	CFS IE2	CFS IE1	CFS IE0	NFS IE3	NFS IE2	NFS IE1	NFS IE0
r						rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Field	Bits	Type	Description
NFSIEx (x = 0-3)	x	rw	Normal Frame Sent in Pipe x Interrupt Enable If set to one the interrupt NFSIx can be generated.
CFSIEx (x = 0-3)	x+4	rw	Command Frame Sent in Pipe x Interrupt Enable If set to one the interrupt CFSIx can be generated.
PEIE	8	rw	Parity Error Interrupt Enable If set to one the interrupt PEI can be generated.
TEIE	9	rw	Time Out Error Interrupt Enable If set to one the interrupt TEI can be generated.
NFSIRx (x = 0-3)	x+16	w	Normal Frame Sent in Pipe x Flag Reset Writing this bit with 1 resets bit TISR.NFSIx . Writing a 0 has no effect. A read action always delivers 0.
CFSIRx (x = 0-3)	x+20	w	Command Frame Sent in Pipe x Flag Reset Writing this bit with 1 resets bit TISR.CFSIx . Writing a 0 has no effect. A read action always delivers 0.
PEIR	24	w	Parity or Time Out Error Flag Reset Writing this bit with 1 resets bit TISR.PEI . Writing a 0 has no effect. A read action always delivers 0.
TEIR	25	w	Time Out Error Flag Reset Writing this bit with 1 resets bit TISR.TEI . Writing a 0 has no effect. A read action always delivers 0.

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
0	[15:10], [31:26]	r	Reserved ; returns 0 if read; should be written with 0.

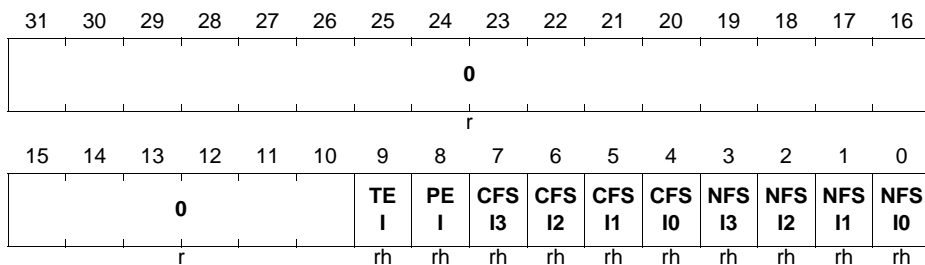
Micro Link Serial Bus Interface (MLI)

7.3.6.2 Transmitter Interrupt Status Register

TISR

Transmitter Interrupt Status Register (9C_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
NFSIx (x = 0-3)	x	rh	Normal Frame Sent in Pipe x Flag It is set to one when a normal frame has been sent out correctly on pipe x.
CFSIx (x = 0-3)	x+4	rh	Command Frame Sent in Pipe x Flag It is set to one when a command frame has been sent out correctly on pipe x.
PEI	8	rh	Parity Error Flag It is set to one when the parity error counter on transmitter side has reached 0.
TEI	9	rh	Time Out Error Flag It is set to one when the non acknowledge counter has reached 0 (it is counting down by 1 with each retry due to READY = 1 when VALID becomes 0).
0	[31:10]	r	Reserved ; returns 0 if read; should be written with 0.

7.3.6.3 Transmitter Interrupt Node Pointer Register

TINPR

Transmitter Interrupt Node Pointer Register (A0_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
0									PTEIP			0	CFSIP			
r									rw			r	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0		NFSIP3			0		NFSIP2		0		NFSIP1		0		NFSIP0	
r		rw			r		rw		r		rw		r		rw	

Field	Bits	Type	Description
NFSIP0	[2:0]	rw	Normal Frame Sent in Pipe 0 Interrupt Pointer Number of the interrupt output reporting the sending of a normal frame through pipe 0 if enabled by NFSIE0 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
NFSIP1	[6:4]	rw	Normal Frame Sent in Pipe 1 Interrupt Pointer Number of the interrupt output reporting the sending of a normal frame through pipe 1 if enabled by NFSIE1 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
NFSIP2	[10:8]	rw	Normal Frame Sent in Pipe 2 Interrupt Pointer Number of the interrupt output reporting the sending of a normal frame through pipe 2 if enabled by NFSIE2 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
NFSIP3	[14:12]	rw	Normal Frame Sent in Pipe 3 Interrupt Pointer Number of the interrupt output reporting the sending of a normal frame through pipe 3 if enabled by NFSIE3 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
CFSIP	[18:16]	rw	Command Frame Sent Interrupt Pointer Number of the common interrupt output reporting the sending of a command frame for: pipe 0 if enabled by CFSIE0 = 1 or Pipe 1 if enabled by CFSIE1 = 1 or Pipe 2 if enabled by CFSIE2 = 1 or Pipe 3 if enabled by CFSIE3 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
PTEIP	[22:20]	rw	Parity or Time Out Interrupt Pointer Number of the common interrupt output reporting: the parity error if enabled by PEIE = 1 or the time out interrupt if enabled by TEIE = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
0	3, 7, 11, 15, 19, [31:23]	r	Reserved ; returns 0 if read; should be written with 0.

7.3.6.4 Receiver Interrupt Enable Register

RIER

Receiver Interrupt Enable Register

(A4_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			0			DRA IR	0	PE IR	ICE R	CFR IR3	CFR IR2	CFR IR1	CFR IR0	ME IR	NFR IR
			r			w	r	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			0			DRA IE	0	PEIE	ICE	CFR IE3	CFR IE2	CFR IE1	CFR IE0		NFR IE
			r			rw	r	rw	rw	rw	rw	rw	rw		rw

Field	Bits	Type	Description
NFRIE	[1:0]	rw	Normal Frame Received Interrupt Enable This bit field defines if an interrupt is generated when a normal frame is correctly received. 00 _B The interrupt generation is disabled. 01 _B The interrupt is generated each time a normal frame is correctly received. 10 _B The interrupt is generated each time a normal frame is correctly received that is not automatically handled by the MLI move engine. 11 _B reserved
CFRIEx (x = 0-3)	x+2	rw	Command Received through Pipe x Interrupt Enable This bit defines if an interrupt is generated when a command frame is correctly received through pipe x. 0 The interrupt generation is disabled. 1 The interrupt generated is enabled.
ICE	6	rw	Interrupt Command Enable This bit defines if an interrupt is generated when a command on pipe 0 requests the interrupt generation. The activated interrupt output line is defined by the command. 0 The interrupt generation is disabled. 1 The interrupt generated is enabled.

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
PEIE	7	rw	Parity Error Interrupt Enable This bit enables the interrupt is generated when a parity error is detected and the parity error counter is 0. 0 The interrupt generation is disabled. 1 The interrupt generation is enabled.
DRAIE	9	rw	Discarded Read Answer Interrupt Enable If set to one the interrupt DRAI can be generated.
NFRIR	16	w	Normal Frame Received Interrupt Flag Reset Writing this bit with 1 resets bit RISR.NFRI . Writing a 0 has no effect. A read action always delivers 0.
MEIR	17	w	MLI Move Engine Interrupt Flag Reset Writing this bit with 1 resets bit RISR.MEI . Writing a 0 has no effect. A read action always delivers 0.
CFRIRx (x = 0-3)	x+18	w	Command Frame Received through pipe x Interrupt Flag Reset Writing this bit with 1 resets bit RISR.CFRIx . Writing a 0 has no effect. A read action always delivers 0.
ICER	22	w	Interrupt Command Flag Reset Writing this bit with 1 resets bit RISR.ICE . Writing a 0 has no effect. A read action always delivers 0.
PEIR	23	w	Parity Error Interrupt Flag Reset Writing this bit with 1 resets bit RISR.PEI . Writing a 0 has no effect. A read action always delivers 0.
DRAIR	25	w	Discarded Read Answer Interrupt Flag Reset Writing this bit with 1 resets bit RISR.DRAI . Writing a 0 has no effect. A read action always delivers 0.
0	8, [15:10], 24, [31:26]	r	Reserved ; returns 0 if read; should be written with 0.

Micro Link Serial Bus Interface (MLI)

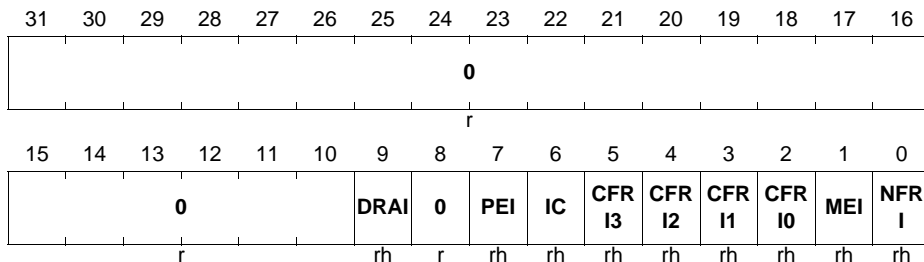
7.3.6.5 Receiver Interrupt Status Register

RISR

Receiver Interrupt Status Register

(A8_H)

Reset Value: 0000 0000_H



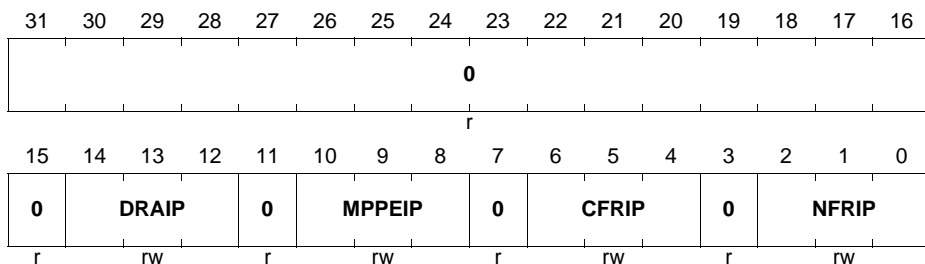
Field	Bits	Type	Description
NFRI	0	rh	Normal Frame Received Interrupt Flag It is set to 1 when received a normal frame.
MEI	1	rh	MLI Move Engine Interrupt Flag It is set when the MLI move engine has done the requested transfer.
CFR _{Ix} (x = 0-3)	x+2	rh	Command Frame Received through pipe x Interrupt Flag It is set to 1 when a command frame has been received correctly on pipe x.
IC	6	rh	Interrupt Command Flag It is set to 1 when an interrupt command on pipe 0 requests an interrupt.
PEI	7	rh	Parity Error Interrupt Flag It is set to one when the parity error counter on receiver side has reached 0.
DRAI	9	rh	Discarded Read Answer Interrupt Flag It is set to one when the answer to a read command has been discarded, because no read pending flag was set.
0	8, [31:10]	r	Reserved ; returns 0 if read; should be written with 0.

7.3.6.6 Receiver Interrupt Node Pointer Register

RINPR

Receiver Interrupt Node Pointer Register (AC_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
NFRIP	[2:0]	rw	Normal Frame Received Interrupt Pointer Number of the interrupt output reporting the reception of a normal frame if enabled by NFRIE. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
CFRIP	[6:4]	rw	Command Frame Received Interrupt Pointer Number of the common interrupt output reporting the reception of a command frame through: pipe 0 if enabled by CFRIE0 = 1 or Pipe 1 if enabled by CFRIE1 = 1 or Pipe 2 if enabled by CFRIE2 = 1 or Pipe 3 if enabled by CFRIE3 = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
MPPEIP	[10:8]	rw	Parity Error Interrupt Pointer Number of the interrupt output reporting a parity error on receiver side if enabled by MPPIE = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
DRAIP	[14:12]	rw	Discarded Read Answer Interrupt Pointer Number of the interrupt output reporting that a read answer has been discarded if enabled by DRAIE = 1. 000 _B MLI interrupt output 0 is selected. ... 111 _B MLI interrupt output 7 is selected.
0	3, 7, 11, [31:15]	r	Reserved ; returns 0 if read; should be written with 0.

Micro Link Serial Bus Interface (MLI)

7.3.6.7 Global Interrupt Set Register

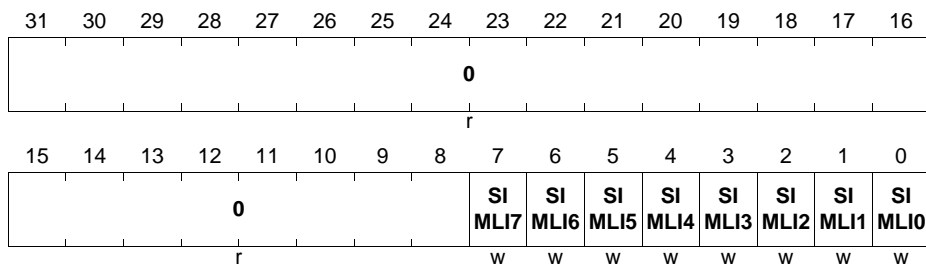
The Global Interrupt Set Register can activate the interrupt output lines of the MLI module. The implementation of this register does not involve any flip flop.

GINTR

Global Interrupt Set Register

(B0_H)

Reset Value: 0000 0000_H



Field	Bits	Type	Description
SIMLIx (x = 0-7)	x	w	Set MLI Interrupt Output Line x 0 No action 1 The MLI interrupt output line x will be activated.
0	[31:8]	r	Reserved ; returns 0 if read; should be written with 0.

7.3.6.8 Output Input Control Register

OICR

Output Input Control Register

(B4_H)

Reset Value: 1000 8000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDP	RDS	RCE	RCP	RCS	RVP	RVS	RRP D	RRP C	RRP B	RRP A	RRS				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RVE	TDP	TCP	TCE	TRE	TRP	TRS	TVP D	TVP C	TVP B	TVP A	TVE D	TVE C	TVE B	TVE A	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
TVEA	0	rw	Transmitter Valid Enable A These bits field enable the transmitter output signals TVALIDA that are driven outside the module. 0 The transmitter output signal TVALIDA is considered as passive. 1 The transmitter output signal TVALIDA reflects the status of the current transmitter kernel signal VALID.
TVEB	1	rw	Transmitter Valid Enable B These bits field enable the transmitter output signals TVALIDB that are driven outside the module. 0 The transmitter output signal TVALIDB is considered as passive. 1 The transmitter output signal TVALIDB reflects the status of the current transmitter kernel signal VALID.
TVEC	2	rw	Transmitter Valid Enable C These bits field enable the transmitter output signals TVALIDC that are driven outside the module. 0 The transmitter output signal TVALIDC is considered as passive. 1 The transmitter output signal TVALIDC reflects the status of the current transmitter kernel signal VALID.

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
TVED	3	rw	Transmitter Valid Enable D These bits field enable the transmitter output signals TVALIDD that are driven outside the module. 0 The transmitter output signal TVALIDD is considered as passive. 1 The transmitter output signal TVALIDD reflects the status of the current transmitter kernel signal VALID.
TVPA	4	rw	Transmitter Valid Polarity A These bits define the polarity of each of the transmitter output signals TVALIDA. 0 An active TVALIDA line is driving a 1, a passive level is 0 (not inverted). 1 An active TVALIDA line is driving a 0, a passive level is 1 (inverted).
TVPB	5	rw	Transmitter Valid Polarity B These bits define the polarity of each of the transmitter output signals TVALIDB. 0 An active TVALIDB line is driving a 1, a passive level is 0 (not inverted). 1 An active TVALIDB line is driving a 0, a passive level is 1 (inverted).
TVPC	6	rw	Transmitter Valid Polarity B These bits define the polarity of each of the transmitter output signals TVALIDB. 0 An active TVALIDB line is driving a 1, a passive level is 0 (not inverted). 1 An active TVALIDB line is driving a 0, a passive level is 1 (inverted).
TVPD	7	rw	Transmitter Valid Polarity D These bits define the polarity of each of the transmitter output signals TVALIDD. 0 An active TVALIDD line is driving a 1, a passive level is 0 (not inverted). 1 An active TVALIDD line is driving a 0, a passive level is 1 (inverted).

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
TRS	[9:8]	rw	Transmitter Ready Selector This bit field defines the transmitter input line that is used for the transmitter kernel signal READY. 00 _B TREADYA is selected. 01 _B TREADYB is selected. 10 _B TREADYC is selected. 11 _B TREADYD is selected.
TRP	10	rw	Transmitter Ready Polarity This bit defines the polarity of the selected transmitter input signal TREADYx. 0 An active TREADYx level is 1, a passive level is 0 (not inverted). 1 An active TREADYx level is 0, a passive level is 1 (inverted).
TRE	11	rw	Transmitter Ready Enable This bit enables the input of the transmitter kernel signal READY. 0 The READY signal is considered as passive (internal = 0). 1 The TREADYx line according to the bit fields TRS and TRP is taken into account.
TCE	12	rw	Transmitter Clock Enable This bit enables the transmitter kernel signal CLK to be driven outside the module. 0 The transmitter clock signal is considered as passive (internal = 0). 1 The TCLK line reflects the status of the current transmitter kernel signal CLK according to bit TCP.
TCP	13	rw	Transmitter Clock Polarity This bit defines the polarity of the transmitter output signal TCLK. 0 A passive TCLK line is driving a 0 (not inverted). 1 A passive TCLK line is driving a 1 (inverted).

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
TDP	14	rw	Transmitter Data Polarity This bit defines the polarity of the transmitter output signal TCLK. 0 The transmitter kernel signal DATA drives directly the transmitter output line TDATA (not inverted). 1 The transmitter kernel signal DATA is inverted before driving the transmitter output line TDATA.
RVE	15	rw	Receiver Valid Enable This bit enables the receiver kernel input signal VALID. 0 The VALID signal is considered as passive (internal = 0). 1 The RVALIDx line according to the bit fields RVS and RVP is taken into account.
RRS	[17:16]	rw	Receiver Ready Selector This bit field defines the receiver output signal RREADYx that is driven outside the module by the receiver kernel signal READY. An RREADYx output signal that is not selected is considered as passive and drives a level according to its corresponding bit RRPx. 00 _B Select RREADYA. 01 _B Select RREADYB. 10 _B Select RREADYC. 11 _B Select RREADYD.
RRPA	18	rw	Receiver Ready Polarity A These bits define the polarity of the receiver output signals RREADYA. 0 An active RREADYA line level is 1, a passive level is 0 (not inverted). 1 An active RREADYA line level is 0, a passive level is 1 (inverted).
RRPB	19	rw	Receiver Ready Polarity B These bits define the polarity of the receiver output signals RREADYB. 0 An active RREADYB line level is 1, a passive level is 0 (not inverted). 1 An active RREADYB line level is 0, a passive level is 1 (inverted).

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
RRPC	20	rw	Receiver Ready Polarity C These bits define the polarity of the receiver output signals RREADYC. 0 An active RREADYC line level is 1, a passive level is 0 (not inverted). 1 An active RREADYC line level is 0, a passive level is 1 (inverted).
RRPA	21	rw	Receiver Ready Polarity D These bits define the polarity of the receiver output signals RREADYD. 0 An active RREADYD line level is 1, a passive level is 0 (not inverted). 1 An active RREADYD line level is 0, a passive level is 1 (inverted).
RVS	[23:22]	rw	Receiver Valid Selector This bit defines which one of the receiver input signals RVALIDx that is taken as input for the receiver kernel signal VALID. 00 _B Select RVALIDA. 01 _B Select RVALIDB. 10 _B Select RVALIDC. 11 _B Select RVALIDD.
RVP	24	rw	Receiver Valid Polarity This bit defines the polarity of the selected receiver input signal RVALIDx. 0 An active RVALIDx level is 1, a passive level is 0 (not inverted). 1 An active RVALIDx level is 0, a passive level is 1 (inverted).
RCS	[26:25]	rw	Receiver Clock Selector This bit defines which one of the receiver input signals RCLKx that is taken as input for the receiver kernel signal CLK. 00 _B Select RCLKA. 01 _B Select RCLKB. 10 _B Select RCLKC. 11 _B Select RCLKD.

Micro Link Serial Bus Interface (MLI)

Field	Bits	Type	Description
RCP	27	rw	Receiver Clock Polarity This bit defines the polarity of the selected receiver input signal RCLKx. 0 An active RCLKx level is 1, a passive level is 0 (not inverted). 1 An active RCLKx level is 0, a passive level is 1 (inverted).
RCE	28	rw	Receiver Clock Enable This bit enables the receiver kernel input signal CLK. 0 The CLK signal is considered as passive (internal = 0). 1 The RCLKx line according to the bit fields RCS and RCP is taken into account.
RDS	[30:29]	rw	Receiver Data Selector This bit defines which one of the receiver input signals RDATAx that is taken as input for the receiver kernel signal DATA. 00 _B Select RDATAA. 01 _B Select RDATAB. 10 _B Select RDATA C. 11 _B Select RDATA D.
RDP	31	rw	Receiver Data Polarity This bit defines the polarity of the selected receiver input signal RDATAx. 0 The receiver kernel signal DATA is directly driven by the selected receiver input line RDATAx (not inverted). 1 The receiver kernel signal DATA is driven by the inverted signal from the selected receiver input line RDATAx.

7.3.6.9 Fractional Divider Registers

The fractional divider register allows the programmer to control the clock rate and period of the 50% duty cycle shift clock f_{MLI} . The period of f_{MLI} can be either $1/STEP$ or a fraction of $STEP/1024$ (for any value of $STEP$ from 0 to 1023) of clock f_{MLI} . Each MLI has its own fractional divider.

Micro Link Serial Bus Interface (MLI)

FDR

MLI Fractional Divider Register

(0C_H)

Reset Value: 03FF 43FF_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIS CLK	0					RESULT									
rwh	r					rh									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DM	0					STEP									
rw	r					rw									

Field	Bits	Type	Description
STEP	[9:0]	rw	Step Value Reload or addition value for RESULT.
DM	[15:14]	rw	Divider Mode This bit field selects normal divider mode or fractional divider mode.
RESULT	[25:16]	rh	Result Value Bit fields for the addition result.
DISCLK	31	rwh	Disable Clock Hardware controlled disable for fOUT signal.
0	[13:10], [30:26]	rw	Reserved ; read as 0; should be written with 0.

7.4 MLI Module Implementation

7.4.1 Interfaces of the MLI Module

7.4.1.1 Port Connections of MLI

Figure 7-51 shows the SAK-CIC310-OSMX2HT specific implementation details and interconnections of the MLI module. The MLI module is further supplied by a separate clock control, interrupt control, address decoding, and port control logic.

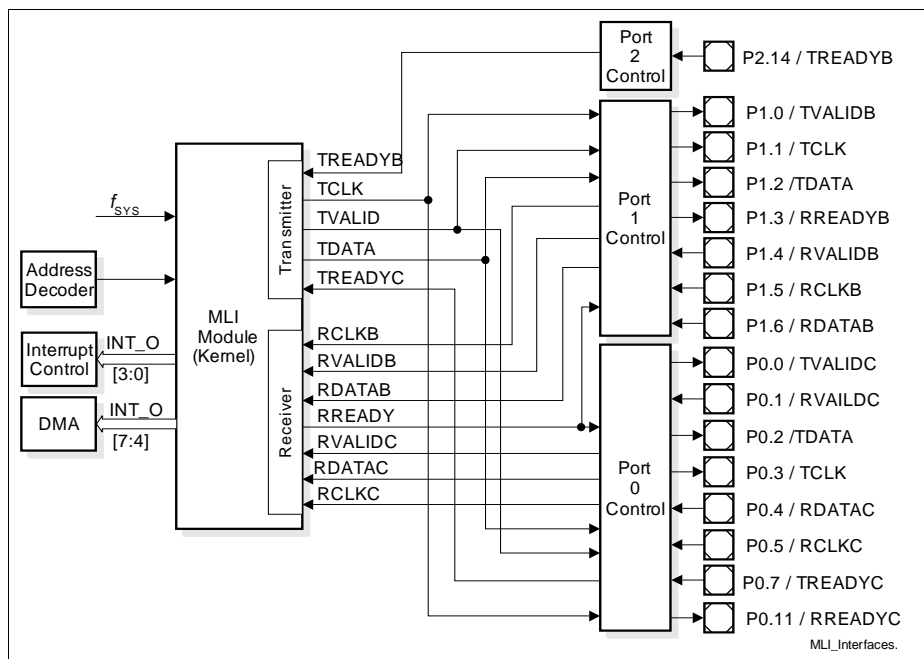


Figure 7-51 MLI Port Connections

7.4.1.2 Interface signals of MLI

The MLI module can be connected to up to 4 other MLI's. For this reason there are 4 sets of interface signals (A, B, C, D), which have to be treated in the way below. For further information see **Figure 7-10**. During the Hardware Reset the MLI signals are switched to setting A. The IME then initializes setting B by programming MLI.OICR. The setting C is reserved for evaluation. The setting D is unused. This means, that

- TVALIDB, TDATA, and TCLK are connected to Port1

Micro Link Serial Bus Interface (MLI)

- TREADYB is connected to Port 2
- TVALIDC, TDATA, TCLK and TREADYC are connected to Port 0
- TVALIDA and TVALIDD remain open
- TREADYA and TREADYD are connected to Vss (Ground)
- RVALIDB, RDATA, RCLKB and RREADYB are connected to Port 1
- RVALIDC, RDATA, RCLKC and RREADYC are connected to Port 0
- RDATAA and RDATA are connected to Vss (Ground)
- RCLKA and RCLKD are connected to Vss (Ground)
- RVALIDA and RVALIDD are connected to Vss (Ground)
- RREADYA and RREADYD remain open

7.4.2 Address Map

In the FlexRay Communication Controller, the registers of the MLI module are located in the following address Range:

- Module Base Address = 0000 0200_H
Module End Address = 0000 02FF_H
- Absolute Register Address = Module Base Address + Offset Address (offset addresses see [Table 7-27](#))

The address ranges for the transfer windows of the MLI module are located in the following address Ranges:

- Small Transfer Windows (8 Kbyte max.):
 - Pipe 0
Pipe Base Address = 0000 8000_H
Module End Address = 0000 9FFF_H
 - Pipe 1
Pipe Base Address = 0000 A000_H
Module End Address = 0000 BFFF_H
 - Pipe 2
Pipe Base Address = 0000 C000_H
Module End Address = 0000 DFFF_H
 - Pipe 3
Pipe Base Address = 0000 E000_H
Module End Address = 0000 FFFF_H
- Large Transfer Windows (64 Kbyte max.):
 - Pipe 0
Pipe Base Address = 0001 0000_H
Module End Address = 0001 FFFF_H
 - Pipe 1
Pipe Base Address = 0002 0000_H
Module End Address = 0002 FFFF_H

Micro Link Serial Bus Interface (MLI)

- Pipe 2
Pipe Base Address = 0003 0000_H
Module End Address = 0003 FFFF_H
- Pipe 3
Pipe Base Address = 0004 0000_H
Module End Address = 0004 FFFF_H

8 External Memory Interface Unit (XMU)

The External Memory Interface Unit (XMU) of the SAK-CIC310-OSMX2HT is the interface between external microcontroller and the internal peripheral units. The basic structure of the XMU is shown in **Figure 8-1**.

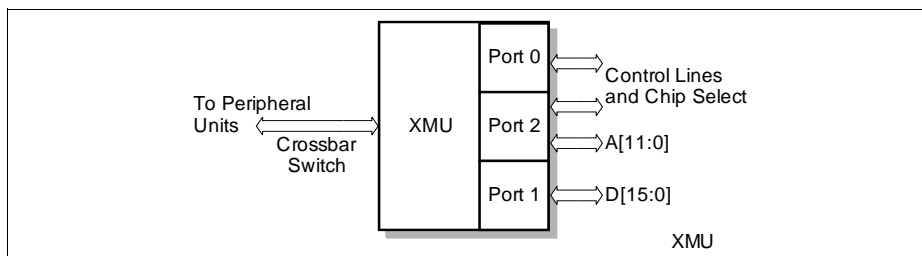


Figure 8-1 XMU Structure and Interfaces

The XMU is primarily used for the communication with external microcontroller. The XMU controls all transactions required for this operation.

8.1 Overview

The XMU enables external bus masters to access all internal on-chip devices connected to the Crossbar switch.

The external bus established by the XMU consists of a 16-bit wide data bus, a 12-bit wide address bus, and a number of control signals.

8.2 XMU Features

- 16-bit wide data bus (D[15:0])
 - Data width of external bus master can be 8or16bits
 - Automatic data assembly/disassembly operation
- 12-bit wide address bus (A[11:0])
- Address extension mechanism to 32-bit
- Bus control signals
 - Read (\overline{RD}) and write (\overline{WR})
 - External synchronous/asynchronous wait state control (\overline{WAIT})
 - External master chip select (\overline{CSFPI}) to access on-chip devices connected to the crossbar switch

External Memory Interface Unit (XMU)

8.3 Basic XMU Operation

The XMU is the gateway from the external world onto the internal on-chip system. **Figure 8-2** shows an example for the connection of an external system, including an external bus master, to the XMU. (Note: not all signals are shown in this diagram)

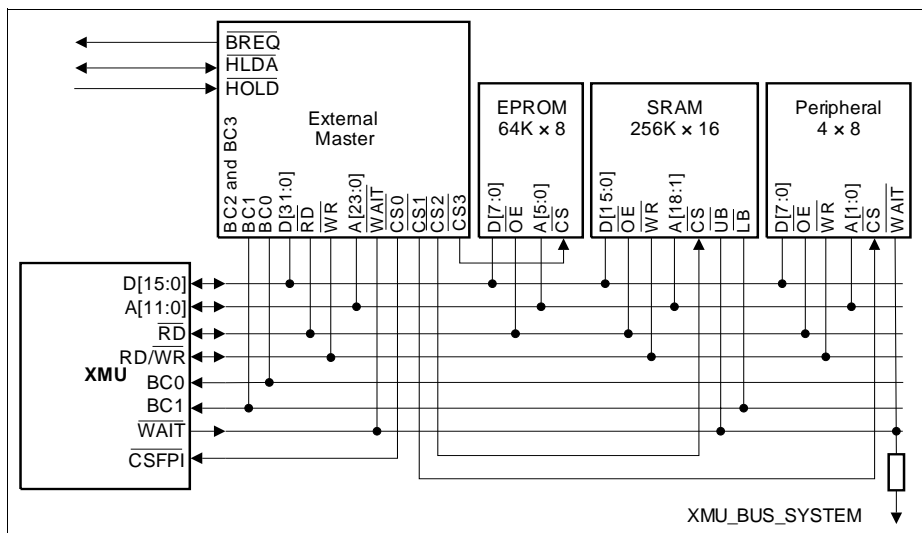


Figure 8-2 Example Configuration for Connection to External Devices

If an external bus master wants to access a device connected to the external bus, it first needs to receive ownership of the external bus.

This external master then performs its transaction over the external bus. If the access is to the internal Crossbar Switch, the XMU is activated as a slave via the chip select signal. It then acts as an bus master on the internal Crossbar Switch, performing the required access on behalf of the external bus master.

8.4 XMU Signal Description

The external signals of the XMU are listed in [Table 8-1](#) and described in the following sections.

Table 8-1 XMU Signals available on the SAK-CIC310-OSMX2HT Ports

Signal	Port	Type	Pull	Function
D[15:0]	Port 1 [15:0]	I/O	–	Data bus lines 0-15
A[11:0]	Port 2 [11:0]	I	–	Address bus lines 0-11
A[12]	Port 0 [8]	I	–	Address bus lines 12
\overline{WR}	Port 2 [13]	I	–	Write control line; active during write operation
\overline{RD}	Port 2 [15]	I	–	Read control line; active during read operation
$\overline{BC0}$	Port 2 [0]multiple xer	I	–	Byte control line n (n =0-1) controls the byte access to corresponding byte location
$\overline{BC1}$	Port 2 [14]	I	–	
\overline{WAIT}	Port 0 [13]	O	–	Wait output
\overline{CSFPI}	Port 2 [12]	I	–	Chip select

8.4.1 Address Bus, A[12:0]

The address bus of the XMU consists of 13 address lines, giving a directly addressable range of 8 KByte. Directly addressable means, that these address lines can be used to access any location within one external device, such as a memory. This external device is selected via the chip select line.

If an external bus master is used in the system, and this master performs an access to the internal bus via the XMU, the address bus is switched to input.

8.4.2 Data Bus, D[16:0]

The Data bus transfers data information in demultiplexed mode, and transfers address and data information in multiplexed mode. The width of this bus is 16 bit. External devices with 8 or 16 bit of data width can be connected to the data bus. The byte control signals, \overline{BCx} , specify which part of the data bus carries valid data. See also [Section 8.4.4](#).

8.4.2.1 Data Bus Direction Control

The Data Bus is driven by the SAK-CIC310-OSMX2HT in case of a Read (low \overline{RD} or active \overline{WR}) or is switched to input (tristate output) in case of a Write (active \overline{RD} or low \overline{WR}). The output signals of the SAK-CIC310-OSMX2HT (D[15:0]), which are inputs of the master, have to be activated by the selected slave and have to be deactivated by all other slaves. In this way it is possible to directly connect the incoming signals at the master's input terminal. The deactivation of the outputs (D[15:0]) of the deselected slaves is accomplished by deactivating the output stage on the associated pads. The activation takes place, if $MODE = 10$ or $MODE=00$ and $\overline{CSFPI} = 0$, i.e. XMU selected.

IOCRB = not (\overline{RD}) and not(\overline{CSFPI}) and not(MODE(0))

Sixteen further bit fields PCxB in register P1_IOCRO, P1_IOCRA, P1_IOCRA, and P1_IOCRA are defined with the same definition PCx as in [Table 9-2](#). If the upper condition is true, the signals IOCRBx become true, then the new bit fields PCxB define the IO port behavior, otherwise the regular PCx defines the GPIO port behavior. These new bit fields are programmable and should be programmed to an input-function in this case. The port thus becomes input, whenever \overline{CSFPI} goes low and the read is selected.

8.4.3 Read and Write Strokes, \overline{RD} and \overline{WR}

Two lines are provided to trigger the read (\overline{RD}) and write (\overline{WR}) operations of external devices. While some read/write devices require only one control input, a RD/ \overline{WR} line, the XMU requires two control input, the read (RD) and write (WR) strobe.

8.4.4 Byte Control Signals, \overline{BCx}

The byte control signals $\overline{BC}[1:0]$ select the appropriate byte lanes of the data bus for both read and write accesses. [Table 8-6](#) and [Table 8-8](#) shows the activation on access from a 16-bit or 8-bit external device. Please note that this scheme supports little-endian devices.

8.4.5 Variable Wait State Control, \overline{WAIT}

This is an output signal to the XMU forcing external masters to insert additional wait states into the access. This signal is driven by the SAK-CIC310-OSMX2HT in case $\overline{CSFPI} = 0$ or is switched to input (tristate output) in case of $\overline{CSFPI} = 1$. The output signals of the SAK-CIC310-OSMX2HT \overline{WAIT} , which are inputs of the master, have to be activated by the selected slave and have to be deactivated by all other slaves. In this way it is possible to directly connect the incoming signals at the master's input terminal. The deactivation of the outputs (\overline{BUSY}) of the deselected slaves is accomplished by deactivating the output stage on the associated pads. The activation takes place, if $MODE = 00$ or $MODE=10$ and $\overline{CSFPI} = 0$, i.e. XMU selected.

IOCRB = not(\overline{CSFPI}) and not(MODE(0))

External Memory Interface Unit (XMU)

One further bit fields PC13B in register P0_IOC12 is defined with the same definition PCx as in [Table 9-2](#). If the upper condition is true, the signals IOCRBx become true, then the new bit field PC13B defines the IO port behavior, otherwise the regular PCx defines the GPIO port behavior. These new bit fields are programmable and should be programmed to an input-function in this case. The port thus becomes input, whenever CSFPI goes low.

8.4.6 XMU Chip Select, $\overline{\text{CSFPI}}$

An external bus master first has to arbitrate for ownership of the external bus. Then, it accesses the external bus with an appropriate address and activates $\overline{\text{CSFPI}}$ to inform the XMU that the access is to be performed from the external bus onto the internal Crossbar Switch. The XMU acts as a slave on the external bus, but as a master on the internal Crossbar Switch. It performs the Crossbar Switch transaction on behalf of the external bus master.

Note: When Port 2 is used as a general purpose I/O port, the function of $\overline{\text{CSFPI}}$ (P2.12) is not disabled. Therefore, $\overline{\text{CSFPI}}$ should be set to high level if its function is not required. If $\overline{\text{CSFPI}}$ = low any falling edge of $\overline{\text{RD}}$ or $\overline{\text{WR}}$ will initiate a slave mode access to the Crossbar Switch.

8.5 Detailed External to Internal XMU Operation

The following subsections provide a more detailed insight into the operation of the XMU for external to internal transactions. Such transactions can be performed by an external bus master that wants to perform a read or write access to an on-chip Crossbar Switch device. The master needs to have ownership of the external bus; thus, bus arbitration will be required in many cases ahead of such an access. This master can access the XMU by activating the XMU chip select input $\overline{\text{CSFPI}}$ and presenting a proper address on the address bus. The features and functions of this operation and its basic timing are described in the following.

8.5.1 XMU Signal Direction

When the XMU is accessed by an external master that wants to read or write an internal module the address bus and control signals are inputs to the XMU, driven by the external master. The data bus is input for writes, and output for reads. The $\overline{\text{WAIT}}$ signal is driven by the XMU to indicate to the master the necessity for additional wait states. [Table 8-2](#) lists the XMU signals for external to internal operation and indicates their direction and relevance for the access.

External Memory Interface Unit (XMU)

Table 8-2 XMU Signals for External to Internal Operation

Signal	Direction	Pull	Driven by
CSFPI	Input	–	External Master
D[15:0]	Input for write access, output for read access	–	External Master for writes, XMU for reads
A[12:0]	Input	–	External Master
RD	Input	–	External Master
WR	Input	–	External Master
BC0	Input	–	External Master
BC1	Input	–	External Master
WAIT/IND	Output (Open Drain)	–	XMU

8.5.2 Address Translation

Because the external address bus is only 24 bits wide, any external address presented by an external bus master must be extended to the full 32-bit Crossbar Switch address width by the XMU. This address extension is performed through three 10-bit extension pointers located inside the XMU. These pointers are accessible only through the external bus with a special addressing scheme.

The three extension pointers provide access to any one of three 4 Mbyte address areas on the Crossbar Switch at any point in time. If other address areas are to be accessed, the pointer values can be reprogrammed by the external bus master. The three extension pointers are located in one 32-bit register, XMU_EXTCON. The upper two bits, A[23:22], of the address presented to the XMU (in external slave mode) select the extension pointer to be used. These two bits are also used to open access to the XMU_EXTCON register for reprogramming, as shown in [Table 8-3](#).

Table 8-3 Selection of the Address Extension Pointers

A[23:22]	Selection of	Location
00 _B	AEXT0	XMU_EXTCON[9:0]
01 _B	Register XMU_EXTCON for reprogramming	–
10 _B	AEXT2	XMU_EXTCON[19:10]
11 _B	AEXT3	XMU_EXTCON[29:20]

External Memory Interface Unit (XMU)

The default values for these extension pointers AEXTn are chosen such that the bottom 4 Mbyte of the three most commonly accessed address segments can be accessed without reprogramming.

Figure 8-3 gives an overview on the address extension operation. The byte-control mapping function generates A[1:0] and specifies the access type from the byte-control signals BC[3:0], which must be generated by the external master. The XMU always behaves as a little-endian 32-bit device that supports aligned accesses only.

Table 8-4 Byte Control Mapping Function

BC3	BC2	BC1	BC0	Crossbar Switch Access Width	Crossbar Switch Address A[1:0]
1	1	1	0	Byte	00 _B
1	1	0	0	Half-word	
0	0	0	0	Word	
1	1	0	1	Byte	01 _B
1	0	1	1	Byte	10 _B
0	0	1	1	Half-word	
0	1	1	1	Byte	11 _B
Other combinations				Undefined	

External Memory Interface Unit (XMU)

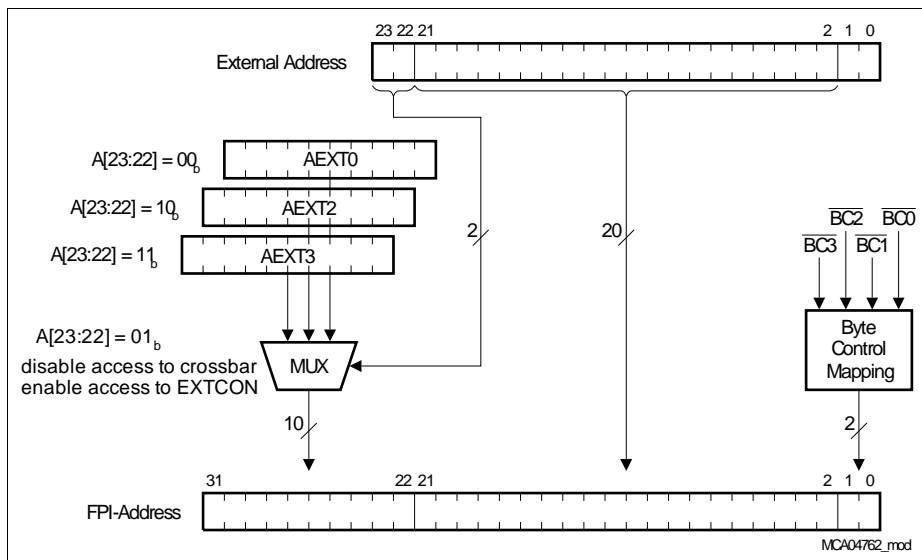


Figure 8-3 External to Internal Address Extension

*Note: It is the user's responsibility to write correct address extension values into register **XMU_EXTCON**. The extended addresses must never point to an external region of memory as this could cause a deadlock on the Crossbar Switch. The XMU would try to access an external region while it is occupied by the access of the external master.*

*Note: If the two most significant external address bits $A[23:22]$ are equal 01_b , no access to the FPI-Bus is initiated (no FPI-Address generated), but instead a XMU Register **XMU_EXTCON** is selected (addressed).*

External Memory Interface Unit (XMU)

8.5.3 External to Internal Access Controls

Two bits in register XMU_CON and one bit in register XMU_EXTCON help to control external accesses to the XMU. Access from an external master to the internal resources via the XMU can be enabled or disabled via bit EXTACC in register XMU_CON. The option to reprogram the address extension registers by the external master can be enabled or disabled through bit ADRC. These controls are useful to prevent hostile or faulty accesses from the external world onto the Crossbar Switch.

8.5.4 Basic Access Timing

When accessed by an external master, the XMU behaves like a 16 bit wide, little-endian device with byte write capability. Accesses must be naturally aligned. Thus, an external master must drive the \overline{BCn} signals and align byte writes to the appropriate data bus byte lane. Usually, the external master derives its clock from a source other than the XMU. Thus, timing synchronization is specified in relation to the \overline{RD} and \overline{WR} signals.

Access time is mainly determined by the time needed for synchronization (two internal clock cycles required, worst case) and the time consumed by the Crossbar Switch transfer (at least TBD internal clock cycles required). The earliest start point of an Crossbar Switch transfer for an external master access is given in [Table 8-5](#).

Table 8-5 Earliest Start Point of Crossbar Switch Transfer for External Master Accesses

Type of Access	Start Point of Crossbar Switch Transfer (earliest)
External master reads from crossbar switch device	Synchronized address bus, \overline{RD} , \overline{WR} and \overline{BC}
External master writes to crossbar switch device	Synchronized address bus, \overline{RD} , \overline{WR} , \overline{BC} and data bus

Read accesses to an Crossbar Switch device therefore require additional wait states, requested through signal \overline{WAIT} by the XMU. Write accesses to the XMU are buffered. A write access will only request additional wait-states through the \overline{WAIT} line if a former write access has not finished yet (consecutive write), else wise the \overline{WAIT} line will remain 1 (first write).

External Memory Interface Unit (XMU)

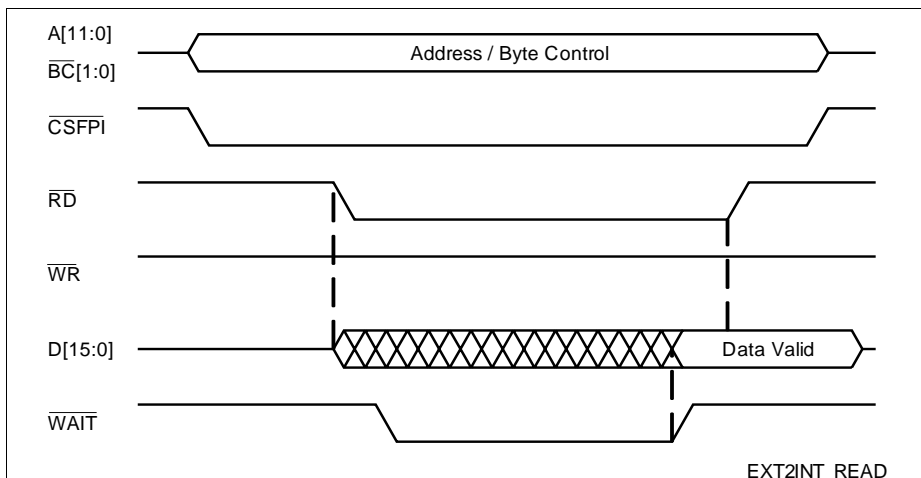


Figure 8-4 Basic External to Internal Read Access Timing

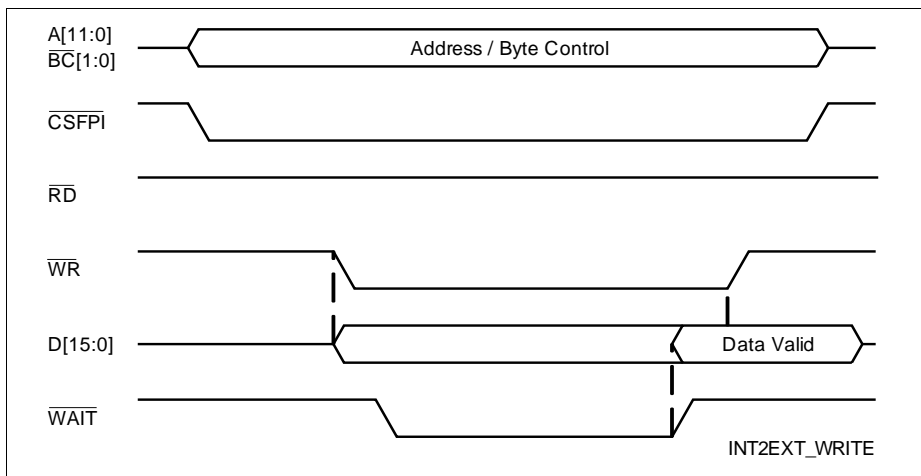


Figure 8-5 Basic External to Internal Write Access Timing (consecutive writes)

External Memory Interface Unit (XMU)

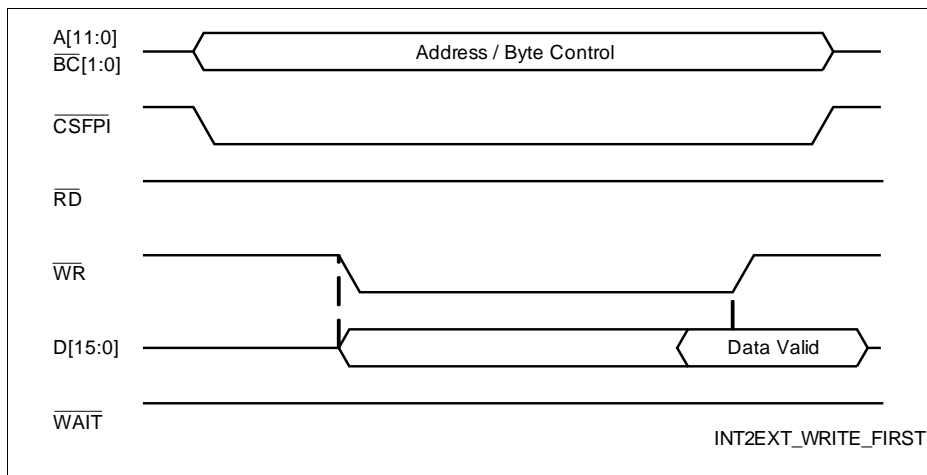


Figure 8-6 Basic External to Internal Write Access Timing (first write)

8.5.5 External Requests to the XMU

If an external bus master requests an internal Crossbar Switch transaction from the XMU to read or write an Crossbar Switch device, this master first has to gain ownership of the external bus. Then it accesses the XMU as a slave device as described in [Section 8.5](#).

The XMU now has to act as a master on the Crossbar Switch on behalf of the external master to fulfill the request. It generates an Crossbar Switch request signal to the Crossbar Switch Control Unit and waits for the bus grant acknowledge. It then takes over the Crossbar Switch and performs the required access.

If the access was a read from an internal device, the XMU asserts the $\overline{\text{WAIT}}$ signal to the external master until it can provide the read data to the master. The external bus master must hold its bus signals active for this time duration.

If the access was a write to an internal device, the write data is stored in a write buffer inside the XMU (provided it is empty, that is, a previous write has finished successfully). The XMU deasserts the $\overline{\text{WAIT}}$ signal as soon as this write buffer store operation has finished. Thus, the external master does need to wait until the internal Crossbar Switch transaction has finished completely. However, if the XMU is performing a previously requested write operation that has not yet finished, the write buffer is not empty, and the external master must wait until the XMU can store the new write data. The XMU asserts $\overline{\text{WAIT}}$ for this time period.

8.5.6 Bidirectional Data Lines

The data lines are driven by the XMU in case of an external read access and driven by the host in case of an external write access. The activation of the outputs (D0 to D15) in case of a external read access is accomplished by activating the output stage of the associated pads. This activation takes place, if $\text{RD} = 0$ and $\text{CSFPI} = 0$, i.e. an external read access.

$$\text{IOCRB} = \overline{\text{RD}} \text{ nor } \overline{\text{CSFPI}}$$

Further bit fields PCxB in registers P1_IOCRO, P1_IOCRO4, P1_IOCRO8, and P1_IOCRO12 are defined with the same definition PCx as in [Table 9-2](#). If the upper condition is true, the signals IOCRxB become true, then the new bit fields PCxB define the GPIO port behavior, otherwise the regular PCx defines the GPIO port behavior. These new bit fields are programmable and should be programmed to an output-function in this case. The port thus becomes output, whenever RD goes low and the chip is selected.

External Memory Interface Unit (XMU)

8.6 XMU Registers

This section describes the control registers and programmable parameters of the XMU. **Figure 8-7** shows all Crossbar Switch accessible registers associated with the XMU. Register XMU_EXTCON, accessible from the external bus only, is described at the end.

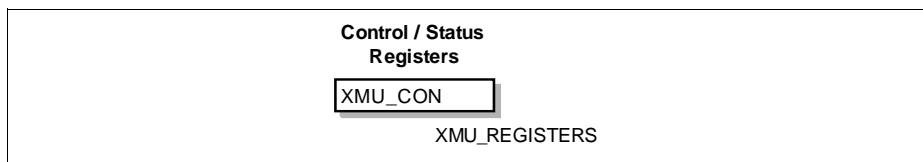


Figure 8-7 XMU Registers

8.6.1 Global Control Register

The XMU Global Control Register XMU_CON provides global control of the XMU. The XMU Global Control Register XMU_CON allows to enable/disable the XMU in general. After reset the XMU is enabled.

XMU_CON

XMU Global Control Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			ADREXT												ADR C
r			rw												rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												BCA0S		EXT ACC	0
r												rw		rw	r

Field	Bits	Type	Description
EXTACC	1	rw	External Access to Crossbar Switch Control 0 External accesses to the Crossbar Switch are disabled (default after reset). 1 External accesses to the Crossbar Switch are enabled.

External Memory Interface Unit (XMU)

Field	Bits	Type	Description
BCA0S	[3:2]	rw	Byte Control and Address 0 Select 00 _B 16 data Lines: XMU $\overline{BC}[1:0]$ is hardwired to 0 (V_{SS}) and a 16 Bit Data Interface is activated 01 _B 16 data lines: XMU $\overline{BC}[0]$ is connected to port pin P2.0, XMU $\overline{BC}[1]$ is connected to port pin P1.7, and a 16 Bit Data Interface is activated 10 _B (reserved) 11 _B 8 data lines: $\overline{BC}[1]$ is unused and a 8 Bit Data Interface is activated ($D[15:8]$ unused)
ADRC	16	rw	External Address Extension Control 0 XMU $A[23:13]$ are hardwired to 0 (V_{SS}). $A[12]$ is connected to port pin P0.8 and $A[11]$ to P2.11. 1 XMU $A[21:11]$ are defined by the bitfield XMU_CON.ADREXT[10:1]. $A[23]$ is connected to port pin P0.8 and $A[22]$ to P2.11. <i>Note: After setting ADRC to 1, XMU_CON may no longer be accessible via XMU. To reset ADRC an internal DMA channels may have to be set up before and triggered by an external event line to reset ADRC respectively.</i>
ADREXT	[28:17]	rw	Address Extension The bitfield ADREWXT[11:1] defines the value of the XMU address lines $A[21:11]$ in case XMU_ADRC is set to 1. <i>Note: The LSB of ADREXT (ADREXT[0]) is only used in JTAG ENABLE Mode.</i>
0	0, [15:4], [31:29]	r	Reserved; read as 0; should be written with 0.

8.6.2 Byte Control Select

The XMU module may transfer 8 bit (byte) and/or 16 bit (half word). 16 bit transfers have to be aligned to halfword addresses ($A[0]=0$). To allow optimization of the pin usage, the

External Memory Interface Unit (XMU)

Byte Control Signals and Address 0 Signal may be connected to a port or alternatively hard wired to a value. This is controlled by the bitfield XMU_CON.BCA0C.

- If using 16 data lines of the XMU, XMU_CON.BCA0S has to be either XMU_CON.BCA0S=00_B or XMU_CON.BCA0S=01_B.
XMU_CON.BCA0S=00_B restricts all XMU operation to be 16 bit (half words) transfers and XMU_CON.BCA0S=01_B enables mixed 8 bit (byte) and/or 16 bit (half word) transfers.
- To use 8 data lines of the XMU only, XMU_CON.BCA0S has to be set to XMU_CON.BCA0S=11_B.
- To use P2.0 as user input, XMU_CON.BCA0S has to be set to XMU_CON.BCA0S=00_B.

XMU_CON is initialized in a manner, enabling users application in any of the above option to be programmed respectively to the application requirements (transfer with only the lower 8 bit sampled by the XMU) Setting of the XMU_CON register therefore has to be one of the first data transfers to be done if using XMU within the application.

External Memory Interface Unit (XMU)

8.7 XMU Module Implementation

8.7.1 Interfaces of the XMU Module

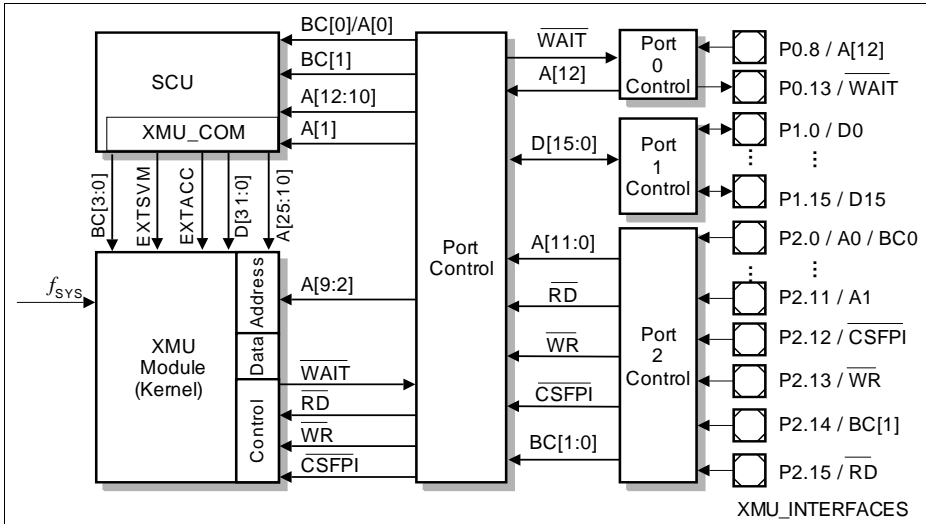


Figure 8-8 XMU Module Implementation and Interconnects

8.7.2 External Address Extension

The XMU module provides 24 address lines. Within the SAK-CIC310-OSMX2HT only 13 address lines A[12:0] are accessible via external pins. The address pins A[23:13] are after reset hardwired to 0 (V_{SS}). This is sufficient to address all registers and E-Ray memory spaces. Only the MLI Large Transfer Windows and Small Transfer Windows (see [Page 7-7](#)) are not accessible. To enable access to these MLI Transfer Windows, an address extension has been implemented. This address extension mechanism is activated by setting the XMU External Address Extension Control bit (XMU_CON.ADRX). But if activating the external address extension, the XMU_CON register may no longer be accessible. Therefore before activating the External Address Extension a DMA channel may be set up to reset the XMU_CON.ADRX bit due to a DMA transfer triggered by an external event line.

If XMU_CON.ADRX is reset (0) and JTAG ENABLE Mode is disabled, the XMU address lines are driven in the following manner:

- A[9:0] by external signals on respective pins of port 2
- A[10] by an external signal on port pin 2.10
- A[11] by an external signal on port pin 2.11

External Memory Interface Unit (XMU)

- A[12] by an external signal on port pin 0.8
- A[23:13] = 0
- The XMU_EXTCONs are not accessible externally.

If XMU_CON.ADRC is set (1) and JTAG ENABLE Mode is disabled, the XMU address lines are driven in the following manner:

- A[9:0] by external signals on respective pins of port 2
- A[10] by an external signal on port pin 2.10
- A[21:11] = XMU_CON_ADREXT[11:1]
- A[22] by an external signal on port pin 2.11
- A[23] by an external signal on port pin 0.8
- The XMU_EXTCONs are accessible externally and may be used to further specify the accessible address range.

If JTAG ENABLE Mode is active, the XMU pin connection is modified as following:

If XMU_CON.ADRC is reset (0), the XMU address lines are driven in the following manner:

- A[9:0] by external signals on respective pins of port 2
- A[10] by an external signal on port pin 2.10
- A[11] by an external signal on port pin 2.11
- A[23:12] = 0
- The XMU_EXTCONs are not accessible externally.

If XMU_CON.ADRC is set (1), the XMU address lines are driven in the following manner:

- A[9:0] by external signals on respective pins of port 2
- A[21:10] = XMU_CON_ADREXT[11:0]
- A[22] by an external signal on port pin 2.10
- A[23] by an external signal on port pin 2.11
- The XMU_EXTCONs are accessible externally and may be used to further specify the accessible address range.

External Memory Interface Unit (XMU)

Table 8-6 Byte Control Mapping for 16-bit Bus (XMU_CON.BCA0S=00_B)

A1	Crossbar Switch Access Width	Crossbar Switch Address A[1:0]
0	Half-word	00 _B
1	Half-word	10 _B
Other combinations	Undefined	

Table 8-7 Byte Control Mapping for 16-bit Bus (XMU_CON.BCA0S=01_B)

A1	BC1	BC0	Crossbar Switch Access Width	Crossbar Switch Address A[1:0]
0	1	0	Byte	00 _B
0	0	0	Half-word	
0	0	1	Byte	01 _B
1	1	0	Byte	10 _B
1	0	0	Half-word	
1	0	1	Byte	11 _B
Other combinations	Undefined			

Table 8-8 Address Mapping for 8-bit Bus (XMU_CON.BCA0S=11_B)

A1	A0	Crossbar Switch Access Width	Crossbar Switch Address A[1:0]
0	0	Byte	00 _B
0	1	Byte	01 _B
1	0	Byte	10 _B
1	1	Byte	11 _B

8.7.3 External Access Configuration Register

The External Access Configuration Register XMU_EXTCONs only accessible from the external bus, not via the internal Crossbar Switch. To reach this register, an external access to the XMU with an address according to the following conditions must be performed:

- A[23:22] = 01_B

External Memory Interface Unit (XMU)

- A[21:3] = don't care
- A[2] = 0
- A[1:0] = don't care.

EXTCON

XMU External Access Configuration Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	AEXT3[9:0]										AEXT2[9:6]			
rw	r	rw										rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AEXT2[5:0]						AEXT0[9:0]									
rw						rw									

Field	Bits	Type	Description
AEXT0	[9:0]	rw	Address Extension Bit Field 0 Specifies address lines A[31:22] for external accesses with A[23:22] = 00 _B .
AEXT2	[19:10]	rw	Address Extension Bit Field 2 Specifies address lines A[31:22] for external accesses with A[23:22] = 10 _B .
AEXT3	[29:20]	rw	Address Extension Bit Field 3 Specifies address lines A[31:22] for external accesses with A[23:22] = 11 _B .
0	30	r	Reserved ; read as 0; should be written with 0.
0	31	rw	Reserved ; reading these bits will return the value last written; read as 0 after reset. Must be written with 0.

Note: A[21:13] are controlled by the XMU_CON register (A[21:12] in JTAG ENABLE Mode)

Note: A[23:22] are either controlled by the external pins or hardwired to 0.

Note: A[12:11] are either controlled by the XMU_CON register or hardwired to 0 (A[11:10] in JTAG ENABLE Mode).

8.8 Supported Address Range

The XMU Interface to the Crossbar Switch only enable an addressable space the XMU can access on the IFLEX of 00000200_H-00001FFF_H. So any XMU external to internal access outside of this window(00000200_H-00001FFF_H) will provide a non defined data in case of an external to internal read and will have no effect in case of an external to internal write.

9.1.1.1 Reset Behavior

During reset, all input stages are enabled and weak pull-ups are activated. Depending on the pin purpose, the behavior can be changed after reset.

9.1.2 Output Driver Control

This section describes the control of the output driver stages.

9.1.2.1 Reset Behavior

During reset, all output stages of GPIO pins go to tristate mode with the weak pull-up enabled.

9.1.2.2 Power-fail Behavior

When the core supply fails while the pad supply remains stable, the output stages go into tristate mode. This is controlled by reset ($\overline{\text{PORST}}$).

9.1.2.3 Pad Driver Control

The pad structure used in this device offers the possibility to select the output driver strength and the slew rate. These selections are independent from the output port functionality, such as open-drain, push/pull, or input only.

In order to minimize EMI problems, the driver strength can be adapted in to the application requirements by the bit fields PDx. The selection is done in groups of four for signals for the normal ports (indicated by PD0, PD1, PD2, and PD3), see [Figure 9-2](#).

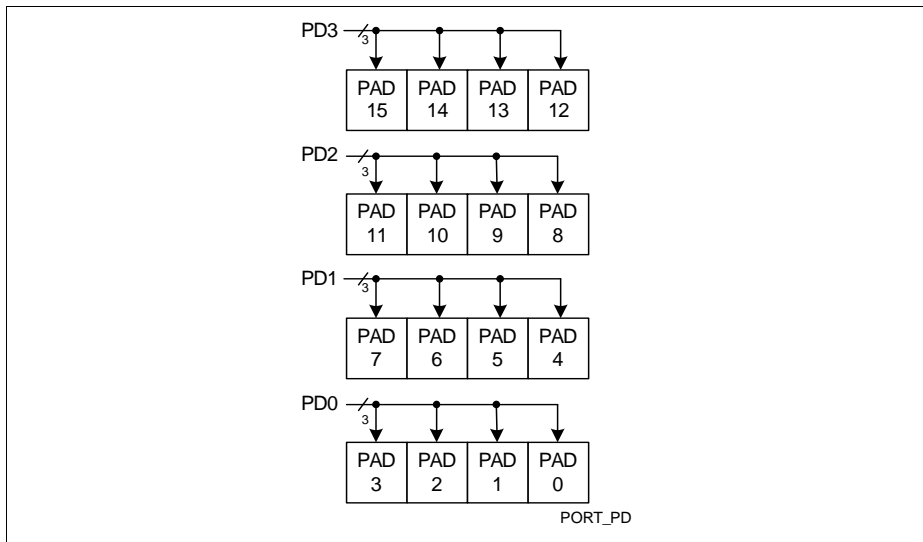


Figure 9-2 Pad Driver Control

A three bit combination of the bit field PDx (pad driver mode) selects between the following possibilities:

Table 9-1 Selection of the Driver Characteristics

PDx.2	PDx.1	PDx.0	Functionality
0	0	0	strong driver, sharp edge ¹⁾
0	0	1	strong driver, medium edge ¹⁾
0	1	0	strong driver, soft edge ¹⁾
0	1	1	undefined
1	0	0	medium driver
1	0	1	medium driver
1	1	0	medium driver
1	1	1	undefined

1) May not be implemented for all ports.

Note: Please refer to the AC characteristics for the exact values.

9.1.2.4 Port Control Coding

The coding of the GPIO port behavior is done by the bit fields in the port control registers Pn_IOC Rx. There's a control bit field PCx for each port pin. The bit fields PCx are located in separate control registers in order to allow modifying a port pin (without influencing the others) with simple move operations.

Table 9-2 PCx Coding

PCx[3:0]	I/O	Output Characteristics	Selected Pull-up/down / Selected Output Function
0000 _B ¹⁾	Direct Input	–	No pull device connected
0001 _B			Not Implemented
0010 _B			Pull-up device connected
0011 _B			Not Implemented
0100 _B	Inverted Input	–	Not Implemented
0101 _B			Not Implemented
0110 _B			Not Implemented
0111 _B			Not Implemented
1000 _B	Output (Direct Input)	Push-pull	General purpose Output
1001 _B			Output function ALT1
1010 _B			Output function ALT2
1011 _B			Output function ALT3
1100 _B		Open Drain	Not Implemented
1101 _B			Not Implemented
1110 _B			Not Implemented
1111 _B			Not Implemented

1) This bit field value is default after reset.

Note: The features listed above are not necessarily implemented in all devices. Please refer to the specific port implementation chapter for more details.

9.1.3 Port Register Description

The bit positions in the port registers always start right-aligned. For example, a port comprising only 8 pins only uses the bit positions [7:0] of the corresponding register. The remaining bit positions are filled with 0 (r).

The pad driver mode registers may be different for each port. As a result, they are described independently for each port in the corresponding chapter.

9.1.3.1 Port Output Register

The port output register defines the values of the output pins if the pin is used as GPIO output.

Pn_OUT

Port n Output Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
Px (x = 0-15)	x	rwh	Port Output Bit x This bit defines the level at the output pin of port Pn, pin x if the output is selected as GPIO output. 0 _B The output level of Pn.x is 0. 1 _B The output level is Pn.x 1.
0	[31:16]	r	Reserved read as 0; should be written with 0.

9.1.3.2 Port Output Modification Register

The port output modification register contains the bits to individually set, reset or toggle the value of the port n output register.

Pn_OMR

Port n Output Modification Register

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PR 15	PR 14	PR 13	PR 12	PR 11	PR 10	PR 9	PR 8	PR 7	PR 6	PR 5	PR 4	PR 3	PR 2	PR 1	PR 0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PS 15	PS 14	PS 13	PS 12	PS 11	PS 10	PS 9	PS 8	PS 7	PS 6	PS 5	PS 4	PS 3	PS 2	PS 1	PS 0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Field	Bits	Type	Description
PSx (x = 0-15)	x	w	Port Set Bit x Setting this bit sets or toggles the corresponding bit in the port output register Pn_OUT (see Table 9-3). On a read access, this bit returns 0.
PRx (x = 0-15)	x + 16	w	Port Reset Bit x Setting this bit resets or toggles the corresponding bit in the port output register Pn_OUT (see Table 9-3). On a read access, this bit returns 0.

Table 9-3 Function of the Bits PRx and PSx

PRx	PSx	Function
0 or no write access	0 or no write access	Bit Pn_OUT.Px is not changed.
0 or no write access	1	Bit Pn_OUT.Px is set.
1	0 or no write access	Bit Pn_OUT.Px is reset.
1	1	Bit Pn_OUT.Px is toggled.

Note: If a bit position is not written (3 out of 4 byte not targeted by a byte write), the corresponding value is considered as 0. Toggling a bit requires one 32-bit write.

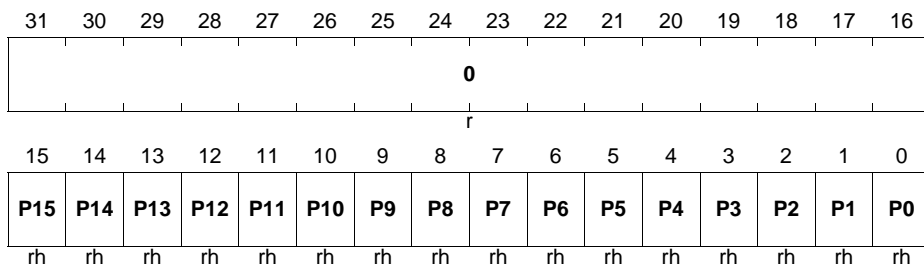
9.1.3.3 Port Input Register

The port input register always contains the values currently read at the pins, independently of the port pin setting (an input or an output).

Pn_IN

Port n Input Register

Reset Value: 0000 XXXX_H¹⁾



1) Px bits for non implemented I/O lines are always read as 0.

Field	Bits	Type	Description
Px (x = 0-15)	x	rh	Port Input Bit x This bit indicates the level at the input pin of port Pn, pin x. 0 _B The input level of Pn.x is 0. 1 _B The input level of Pn.x is 1.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

9.1.3.4 Port Input/Output Control Registers

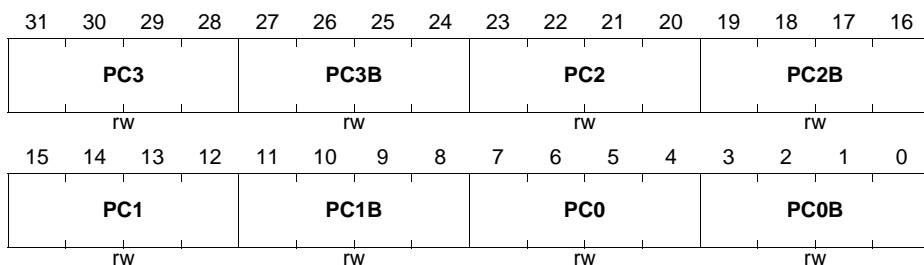
The port input/output control registers contain the bit fields to select the digital output and input driver characteristics, such as pull-up/down devices, port direction (input/output), open-drain and alternate output selections. The coding of the functionality is shown in [Table 9-2](#).

Depending on the port functionality not all of the input/output control registers may be implemented. The structure with one control bit field for each port pin located in different register byte offers the possibility to configure port pin functionality of a single pin without accessing the other PCx by byte-oriented writes.

Pn_IOCRO

Port n Input/Output Control Register 0

Reset Value: 0000 0000_H



Field	Bits	Type	Description
PCx x = 0 x = 1 x = 2 x = 3)	[7:4], [15:12], [23:20], [31:28]	rw	Port Control for Port n Pin x This bit field defines the port functionality according to the coding table (Table 9-2).
PCxB x = 0 x = 1 x = 2 x = 3	[3:0], [11:8], [19:16], [27:24]	rw	Alternative Port Control for Port n Pin x This bit field defines the port functionality according to the coding table (Table 9-2) if selected by IO module by setting IOCRxB signal.

Parallel Ports (PORTS)

Pn_IOC4

Port n Input/Output Control Register 4

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC7				PC7B				PC6				PC6B			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC5				PC5B				PC4				PC4B			
rw				rw				rw				rw			

Field	Bits	Type	Description
PCx x = 4 x = 5 x = 6 x = 7	[7:4], [15:12], [23:20], [31:28]	rw	Port Control for Port n Pin x This bit field defines the port functionality according to the coding table (Table 9-2).
PCxB x = 4 x = 5 x = 6 x = 7	[3:0], [11:8], [19:16], [27:24]	rw	Alternative Port Control for Port n Pin x This bit field defines the port functionality according to the coding table (Table 9-2) if selected by IO module by setting IOCxB signal.

Pn_IOC8

Port n Input/Output Control Register 8

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC11				PC11B				PC10				PC10B			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC9				PC9B				PC8				PC8B			
rw				rw				rw				rw			

Field	Bits	Type	Description
PCx x = 8 x = 9 x = 10 x = 11	[7:4], [15:12], [23:20], [31:28]	rw	Port Control for Port n Pin x This bit field defines the port functionality according to the coding table (Table 9-2).
PCxB x = 8 x = 9 x = 10 x = 11	[3:0], [11:8], [19:16], [27:24]	rw	Alternative Port Control for Port n Pin x This bit field defines the port functionality according to the coding table (Table 9-2) if selected by IO module by setting IOC8xB signal.

Parallel Ports (PORTS)

Pn_IOC12

Port n Input/Output Control Register 12

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC15				PC15B				PC14				PC14B			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC13				PC13B				PC12				PC12B			
rw				rw				rw				rw			

Field	Bits	Type	Description
PCx x = 0, 4, 8, 12 x = 1, 5, 9 x = 2, 6, 10 x = 3, 7, 11)	[7:4], [15:12], [23:20], [31:28]	rw	Port Control for Port n Pin x This bit field defines the port functionality according to the coding table (Table 9-2).
PCx x = 0, 4, 8, 12 x = 1, 5, 9 x = 2, 6, 10 x = 3, 7, 11)	[3:0], [11:8], [19:16], [27:24]	rw	Alternative Port Control for Port n Pin x This bit field defines the port functionality according to the coding table (Table 9-2) if selected by IO module by setting IOC12B signal.

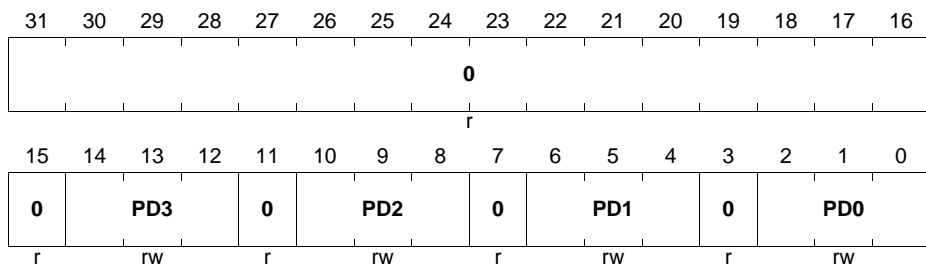
9.1.3.5 Pad Driver Mode Register

The port pad driver mode register contain bit fields (coding see [Table 9-1](#)) that define the output driver strength and slew rate of the port lines.

Pn_PDR

Port n Pad Driver Mode Register

Reset Value: 0000 0000_H



Field	Bits	Type	Description
PD0	[2:0]	rw	Pad Driver Mode for Pn.[3:0] Coding see Table 9-1 .
PD1	[6:4]	rw	Pad Driver Mode for Pn.[7:4] Coding see Table 9-1 .
PD2	[10:8]	rw	Pad Driver Mode for Pn.[11:8] Coding see Table 9-1 .
PD3	[14:12]	rw	Pad Driver Mode for Pn.[15:12] Coding see Table 9-1 .
0	3, 7, 11, [31:15]	r	Reserved ; read as 0; should be written with 0.

9.2 Pin Description

This section summarizes all implementation details of the ports.

9.2.1 Device Pinning Overview

The SAK-CIC310-OSMX2HT has the following I/O ports:

Table 9-4 Ports of the SAK-CIC310-OSMX2HT

Group	Width	I/O	Used for
P0	14	I/O, I	MLI, E-Ray, SCU
P1	16	I/O	SSC, XMU, MLI, SCU
P2	16	I	MLI, XMU

The the input characteristics (with pull-up, pull-down) are hard wired during design and cannot be programmed by the user in the field.

9.2.2 Port 0 Registers

The following section describes all the implementation details on port 1.

9.2.2.1 Overview

Port 0 is a general purpose 14-bit bidirectional port, with exception of pin P0.4-P0.7 only being input. The port registers of Port 0 are shown in [Figure 9-3](#).

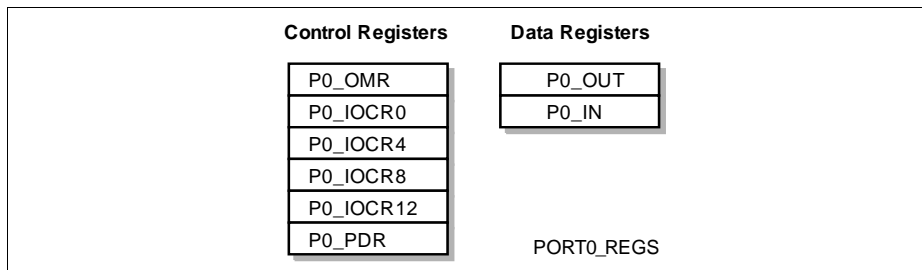


Figure 9-3 Port 0 Register Overview

Parallel Ports (PORTS)

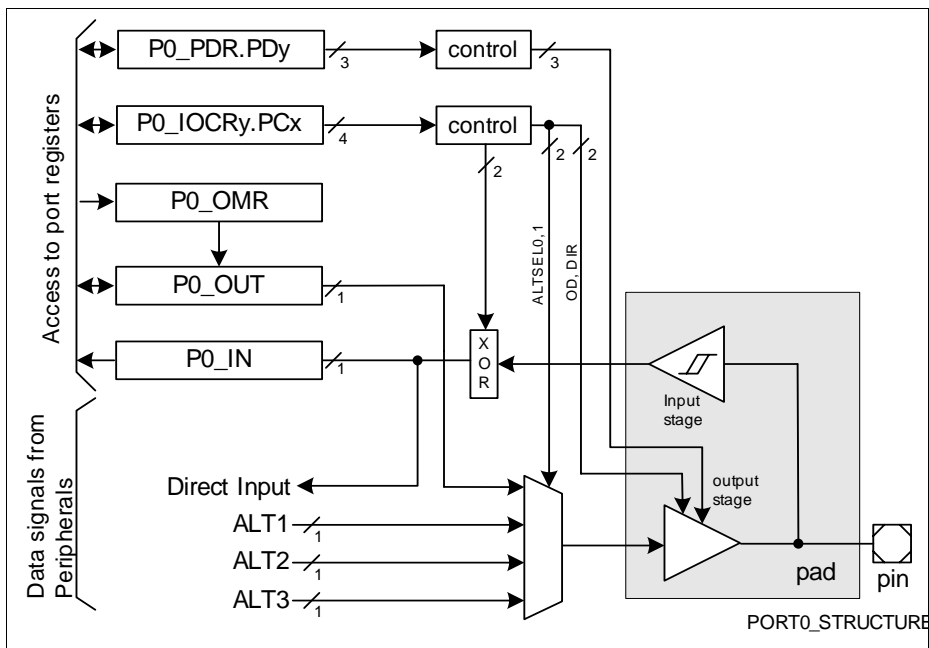


Figure 9-4 Port 0 Structure

9.2.2.2 Port 0 Register Implementations

Table 9-5 Registers Address SpacePort_0 Kernel Register Address Space

Module	Base Address	End Address	Note
P0	0000 0A00 _H	0000 0AFF _H	256 byte P0-P2 have Common Address Space

Table 9-6 Registers OverviewPort 0 Registers

Register Short Name	Register Long Name	Relative Address ¹⁾	Description see
P0_OUT	Port 0 Output Register	00 _H	Page 9-16
P0_OMR	Port 0 Output Modification Register	04 _H	Page 9-17
	Reserved	08 _H	
	Reserved	0C _H	
P0_IOCRO	Port 0 Input/Output Control Register 0	10 _H	Page 9-18
P0_IOCRO4	Port 0 Input/Output Control Register 4	14 _H	Page 9-20
P0_IOCRO8	Port 0 Input/Output Control Register 8	18 _H	Page 9-22
P0_IOCRO12	Port 0 Input/Output Control Register 12	1C _H	Page 9-23
	Reserved	20 _H	
P0_IN	Port 0 Input Register	24 _H	Page 9-24
P0_PDR	Port 0 Pad Driver Mode Register	28 _H	Page 9-25
	Reserved	2C _H	

1) The relative address indicates the offset to the base address.

Parallel Ports (PORTS)

P0_OUT

Port 0 Output Register

(00_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	P13	P12	P11	P10	P9	P8	0					P3	P2	P1	P0
r	rw	rwh	rwh	rwh	rwh	rwh	r					rwh	rwh	rwh	rwh

Field	Bits	Type	Description
Px (x = 0-3)	x	rwh	Port Output Bit x This bit defines the level at the output pin of port P0, pin x if the output is selected as GPIO output. 0 _B The output level of P0.x is 0. 1 _B The output level is P0.x 1.
Px (x = 8-13)	x	rwh	Port Output Bit x This bit defines the level at the output pin of port P0, pin x if the output is selected as GPIO output. 0 _B The output level of P0.x is 0. 1 _B The output level is P0.x 1.
0	[7:4], [31:14]	r	Reserved ; read as 0; should be written with 0.

Parallel Ports (PORTS)

P0_OMR

Port 0 Output Modification Register (04_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		PR 13	PR 12	PR 11	PR 10	PR 9	PR 8		0			PR 3	PR 2	PR 1	PR 0
r		w	w	w	w	w	w		r			w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		PS 13	PS 12	PS 11	PS 10	PS 9	PS 8		0			PS 3	PS 2	PS 1	PS 0
r		w	w	w	w	w	w		r			w	w	w	w

Field	Bits	Type	Description
PSx (x = 0-3)	x	w	Port Set Bit x Setting this bit sets or toggles the corresponding bit in the port output register P0_OUT (see Table 9-3). On a read access, this bit returns 0.
PSx (x = 8-13)	x	w	Port Set Bit x Setting this bit sets or toggles the corresponding bit in the port output register P0_OUT (see Table 9-3). On a read access, this bit returns 0.
PRx (x = 0-3)	x + 16	w	Port Reset Bit x Setting this bit resets or toggles the corresponding bit in the port output register P0_OUT (see Table 9-3). On a read access, this bit returns 0.
PRx (x = 8-13)	x + 16	w	Port Reset Bit x Setting this bit resets or toggles the corresponding bit in the port output register P0_OUT (see Table 9-3). On a read access, this bit returns 0.
0	[7:4], [23:20], [15:14], [31:30]	r	Reserved ; read as 0; should be written with 0.

Parallel Ports (PORTS)

P0_IOCRO

Port 0 Input/Output Control Register 0(10_H)

Reset Value: 2020 9090_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC3				0				PC2				0			
rw				r				rw				r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC1				0				PC0				0			
rw				r				rw				r			

Field	Bits	Type	Description
PC0	[7:4]	rw	Port Control for Port 0 Pin 0 This bit field defines the port functionality according to the coding table (Table 9-7).
PC1	[15:12]	rw	Port Control for Port 0 Pin 1 This bit field defines the port functionality according to the coding table (Table 9-7).
PC2	[23:20]	rw	Port Control for Port 0 Pin 2 This bit field defines the port functionality according to the coding table (Table 9-7).
PC3	[31:28]	rw	Port Control for Port 0 Pin 3 This bit field defines the port functionality according to the coding table (Table 9-7).
0	[3:0], [11:8], [19:16], [27:24]	r	Reserved ; read as 0; should be written with 0.

Table 9-7 PCx Coding for Port 0

PCx[3:0]	I/O	Output Characteristics	Selected Pull-up/down / Selected Output Function
0000 _B ¹⁾	Direct Input	—	No pull device connected
0001 _B			Not Implemented
0010 _B			Pull-up device connected
0011 _B			Not Implemented
0100 _B	Inverted Input	—	Not Implemented
0101 _B			Not Implemented
0110 _B			Not Implemented
0111 _B			Not Implemented
1000 _B	Output (Direct Input)	Push-pull	General purpose Output
1001 _B			Output function ALT1
1010 _B			Output function ALT2
1011 _B			Output function ALT3
1100 _B		Open Drain	Not Implemented
1101 _B			Not Implemented
1110 _B			Not Implemented
1111 _B			Not Implemented

1) This bit field value is default after reset.

P0_IOCRA

Port 0 Input/Output Control Register 4(14_H)

Reset Value: 2020 2020_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC7				0				PC6				0			
rw				r				rw				r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC5				0				PC4				0			
rw				r				rw				r			

Field	Bits	Type	Description
PC4	[7:4]	rw	Port Control for Port 0 Pin 4 This bit field defines the port functionality according to the coding table (Table 9-8).
PC5	[15:12]	rw	Port Control for Port 0 Pin 5 This bit field defines the port functionality according to the coding table (Table 9-8).
PC6	[23:20]	rw	Port Control for Port 0 Pin 6 This bit field defines the port functionality according to the coding table (Table 9-8).
PC7	[31:28]	rw	Port Control for Port 0 Pin 7 This bit field defines the port functionality according to the coding table (Table 9-8).
0	[3:0], [11:8], [19:16], [27:24]	r	Reserved ; read as 0; should be written with 0.

Table 9-8 PC4-PC7 Coding for Port 0

PCx[3:0]	I/O	Output Characteristics	Selected Pull-up/down / Selected Output Function
0000 _B ¹⁾	Direct Input	—	No pull device connected
0001 _B			Not Implemented
0010 _B			Pull-up device connected
0011 _B			Not Implemented
0100 _B	Inverted Input	—	Not Implemented
0101 _B			Not Implemented
0110 _B			Not Implemented
0111 _B			Not Implemented
1000 _B	Output (Direct Input)	Push-pull	Not Implemented
1001 _B			Not Implemented
1010 _B			Not Implemented
1011 _B			Not Implemented
1100 _B		Open Drain	Not Implemented
1101 _B			Not Implemented
1110 _B			Not Implemented
1111 _B			Not Implemented

1) This bit field value is default after reset.

P0_IOC8

Port 0 Input/Output Control Register 8(18_H)

Reset Value: 2020 2020_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC11				0				PC10				0			
rw				r				rw				r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC9				0				PC8				0			
rw				r				rw				r			

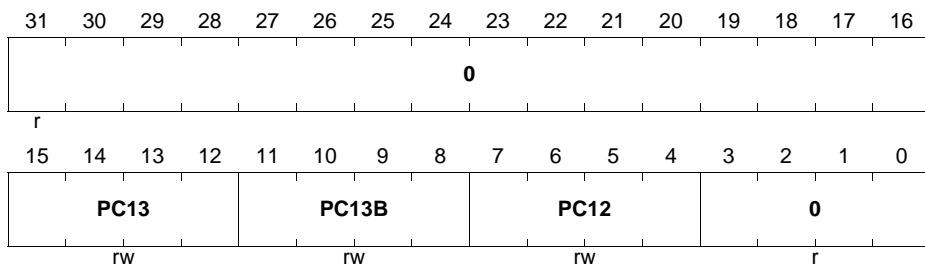
Field	Bits	Type	Description
PC8	[7:4]	rw	Port Control for Port 0 Pin 8 This bit field defines the port functionality according to the coding table (Table 9-7).
PC9	[15:12]	rw	Port Control for Port 0 Pin 9 This bit field defines the port functionality according to the coding table (Table 9-7).
PC10	[23:20]	rw	Port Control for Port 0 Pin 10 This bit field defines the port functionality according to the coding table (Table 9-7).
PC11	[31:28]	rw	Port Control for Port 0 Pin 11 This bit field defines the port functionality according to the coding table (Table 9-7).
0	[3:0], [11:8], [19:16], [27:24]	r	Reserved ; read as 0; should be written with 0.

Parallel Ports (PORTS)

P0_IOC12

Port 0 Input/Output Control Register 12(1C_H)

Reset Value: 0000 2220_H



Field	Bits	Type	Description
PC12	[7:4]	rw	Port Control for Port 0 Pin 12 This bit field defines the port functionality according to the coding table (Table 9-7).
PC13B	[11:8]	rw	Alternative Port Control for Port 0 Pin 13 This bit field defines the port functionality according to the coding table (Table 9-7) if selected by IO module by setting IOC13B signal.
PC13	[15:12]	rw	Port Control for Port 0 Pin 13 This bit field defines the port functionality according to the coding table (Table 9-7).
0	[3:0], [31:16]	r	Reserved ; read as 0; should be written with 0.

Parallel Ports (PORTS)

P0_IN

Port 0 Input Register

(24_H)

Reset Value: 0000 XXXX_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0	
r	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
Px (x = 0-13)	x	rh	Port Input Bit x This bit indicates the level at the input pin of port P0, pin x. 0 The input level of P0.x is 0. 1 The input level of P0.x is 1.
0	[31:14]	r	Reserved ; read as 0; should be written with 0.

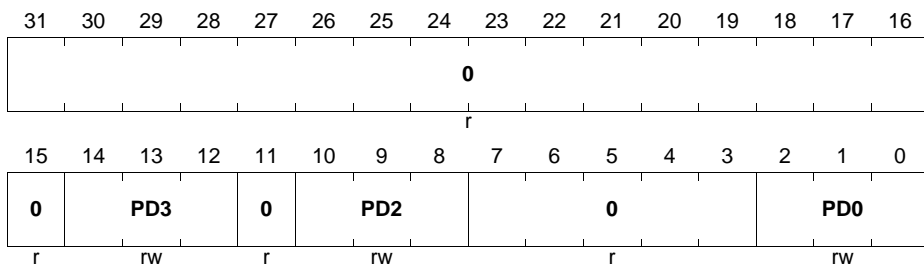
Parallel Ports (PORTS)

P0_PDR

Port 0 Pad Driver Mode Register

(28_H)

Reset Value: 0000 5500_H



Field	Bits	Type	Description
PD0	[2:0]	rw	Pad Driver Mode for P0.[3:0] Coding see Table 9-1 .
PD2	[10:8]	rw	Pad Driver Mode for P0.[11:8] Coding see Table 9-9 .
PD3	[14:12]	rw	Pad Driver Mode for P0.[15:12] Coding see Table 9-9 .
0	[7:3], 11, [31:15]	r	Reserved ; read as 0; should be written with 0.

Table 9-9 Selection of the Driver Characteristics of Port0.[15:8]

PDx.2	PDx.1	PDx.0	Functionality
0	0	0	undefined
0	0	1	undefined
0	1	0	undefined
0	1	1	undefined
1	0	0	medium driver
1	0	1	medium driver
1	1	0	medium driver
1	1	1	undefined

9.2.2.3 Port 0 Functions

The following table summarizes the signals routed to the Port 0 pads.

Table 9-10 Port 0 Input/Output Functions

Port Pin	I/O	Select	Connected Signal(s)	From / to Module
P0.0	Input		Port Input Register P0_IN.0	Port Input
	Output	GPIO	Port Output Register P0_OUT.0	Port Output
		ALT1	$\overline{\text{TXENB}}$	E-Ray
		ALT2	Not Implemented	
		ALT3	TVALIDC	MLI
		JTAG	TDO ¹⁾	JTAG
P0.1	Input		Port Input Register P0_IN.1, RVALIDC, TCK ¹⁾	Port Input, MLI, JTAG
	Output	GPIO	Port Output Register P0_OUT.1	Port Output
		ALT1	$\overline{\text{TXENA}}$	E-Ray
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P0.2	Input		Port Input Register P0_IN.2	Port Input
	Output	GPIO	Port Output Register P0_OUT.2	Port Output
		ALT1	TXDB	E-Ray
		ALT2	Not Implemented	
		ALT3	TDATA	MLI
P0.3	Input		Port Input Register P0_IN.3	Port Input
	Output	GPIO	Port Output Register P0_OUT.3	
		ALT1	TXDA	E-Ray
		ALT2	Not Implemented	
		ALT3	TCLK	MLI

Parallel Ports (PORTS)
Table 9-10 Port 0 Input/Output Functions (cont'd)

Port Pin	I/O	Select	Connected Signal(s)	From / to Module
P0.4	Input		Port Input Register P0_IN.4, RXDB, RDATA_C	Port Input, E-Ray, MLI
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P0.5	Input		Port Input Register P0_IN.5, RXDA, RCLKC	Port Input, E-Ray, MLI
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P0.6	Input		Port Input Register P0_IN.6, BGEB, SR14	Port Input, E-Ray, SCU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P0.7	Input		Port Input Register P0_IN.7, BGEA, SR13, TREADYC, TMS ¹⁾	Port Input, E-Ray, SCU, MLI, JTAG
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P0.8	Input		Port Input Register P0_IN.8, SR8, $\overline{\text{TRST}}$, A12	Port Input, SCU, XMU
	Output	GPIO	Port Output Register P0_OUT.8	Port Output
		ALT1	INT_O1 OR INT_O2	SCU
		ALT2	INT_O1	SCU
		ALT3	INT_O2	SCU

Parallel Ports (PORTS)
Table 9-10 Port 0 Input/Output Functions (cont'd)

Port Pin	I/O	Select	Connected Signal(s)	From / to Module
P0.9	Input		Port Input Register P0_IN.9, SR12	Port Input, SCU
	Output	GPIO	Port Output Register P0_OUT.9	Port Output
		ALT1	ARM	E-Ray
		ALT2	INT_O0	SCU
		ALT3	INT_O1	SCU
P0.10	Input		Port Input Register P0_IN.10, SR11	Port Input, SCU
	Output	GPIO	Port Output Register P0_OUT.10	Port Output
		ALT1	MT	E-Ray
		ALT2	INT_O2	SCU
		ALT3	INT_O3	SCU
P0.11	Input		Port Input Register P0_IN.11, SR10, TDI	Port Input, SCU, JTAG
	Output	GPIO	Port Output Register P0_OUT.11	Port Output
		ALT1	BGT	E-Ray
		ALT2	INT_O0	SCU
		ALT3	RREADYC	MLI
P0.12	Input		Port Input Register P0_IN.12, SR9, MODE[0]	Port Input, SCU
	Output	GPIO	Port Output Register P0_OUT.12	Port Output
		ALT1	INT_O0 OR INT_O4	SCU
		ALT2	INT_O0	SCU
		ALT3	INT_O4	SCU
P0.13	Input		Port Input Register P0_IN.13	Port Input
	Output	GPIO	Port Output Register P0_OUT.13	Port Output
		ALT1	WAIT	XMU
		ALT2	Not Implemented	
		ALT3	Not Implemented	

1) Only in case of JTAG Enable Mode

9.2.3 Port 1 Registers

Port 1 is a general purpose 16-bit bidirectional port.

9.2.3.1 Overview

The port registers of Port 1 are shown in [Figure 9-5](#).

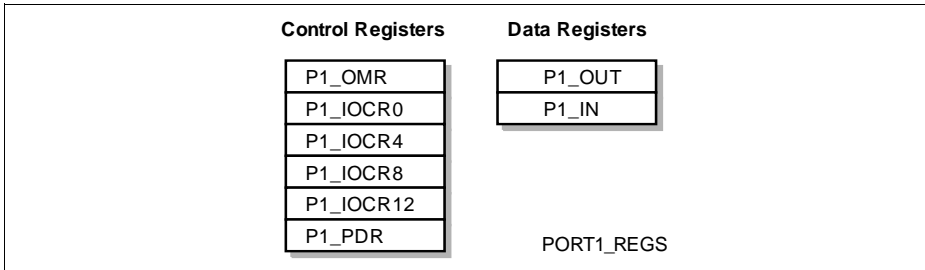


Figure 9-5 Port 1 Register Overview

The definition of all P1_IOCRA supports additional features to handle XMU, and SSC operations. Due to the necessary deactivation of the output stage on all pads of this port the bitfields PCxB have been added to all P1_IOCRA registers. They have exactly the same structure as the other bit fields. PCn and PCnB are only mutually in effect, controlled by a signal IOCRA.

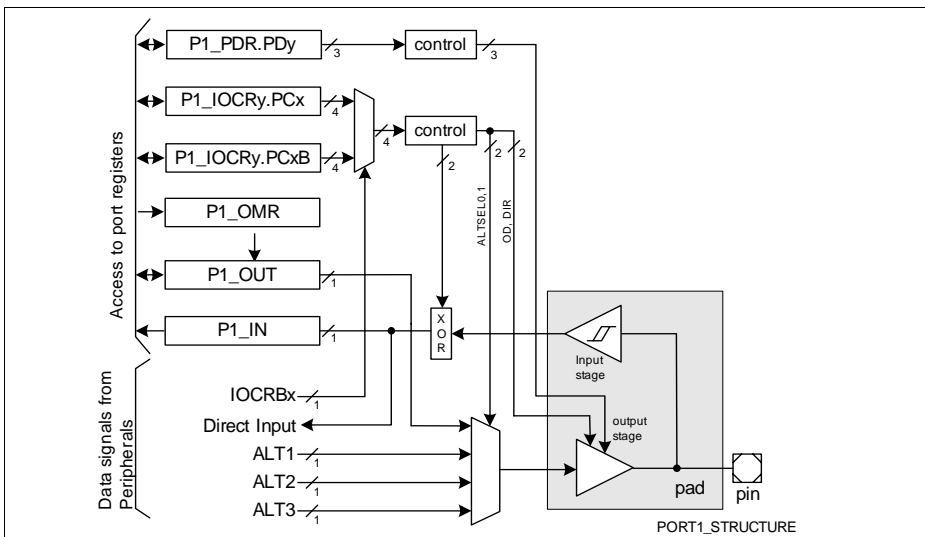


Figure 9-6 Port1 Structure

9.2.3.2 Port 1 Register Implementations

Table 9-11 Registers Address SpacePort_1 Kernel Register Address Space

Module	Base Address	End Address	Note
P1	0000 0A00 _H	0000 0AFF _H	256 byte P0-P2 have Common Address Space

Table 9-12 Port 1 Registers

Register Short Name	Register Long Name	Relative Address ¹⁾	Description see
P1_OUT	Port 1 Output Register	30 _H	Page 9-31
P1_OMR	Port 1 Output Modification Register	34 _H	Page 9-32
	Reserved	38 _H	
	Reserved	3C _H	
P1_IOCRO	Port 1 Input/Output Control Register 0	40 _H	Page 9-33
P1_IOCRO4	Port 1 Input/Output Control Register 4	44 _H	Page 9-35
P1_IOCRO8	Port 1 Input/Output Control Register 8	48 _H	Page 9-36
P1_IOCRO12	Port 1 Input/Output Control Register 12	4C _H	Page 9-37
	Reserved	50 _H	
P1_IN	Port 1 Input Register	54 _H	Page 9-39
P1_PDR	Port 1 Pad Driver Mode Register	58 _H	Page 9-40
	Reserved	5C _H	

1) The relative address indicates the offset to the base address.

Parallel Ports (PORTS)

P1_OUT

Port 1 Output Register

(30_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
Px (x = 0-15)	x	rwh	Port Output Bit x This bit defines the level at the output pin of port P1, pin x if the output is selected as GPIO output. 0 The output level of P1.x is 0. 1 The output level is P1.x 1.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Parallel Ports (PORTS)

P1_OMR

Port 1 Output Modification Register (34_H)

Reset Value: 0000 0000_H

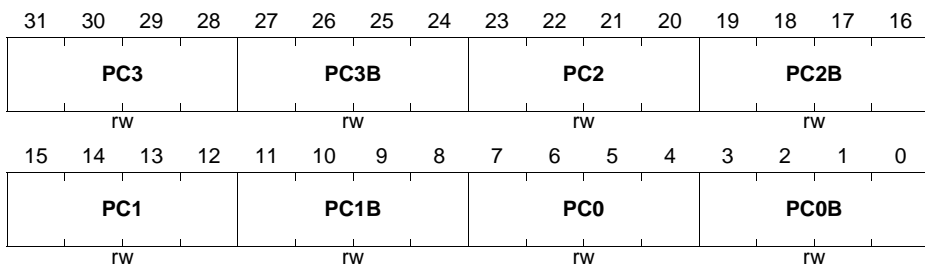
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PR 15	PR 14	PR 13	PR 12	PR 11	PR 10	PR 9	PR 8	PR 7	PR 6	PR 5	PR 4	PR 3	PR 2	PR 1	PR 0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PS 15	PS 14	PS 13	PS 12	PS 11	PS 10	PS 9	PS 8	PS 7	PS 6	PS 5	PS 4	PS 3	PS 2	PS 1	PS 0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Field	Bits	Type	Description
PSx (x = 0-15)	x	w	Port Set Bit x Setting this bit sets or toggles the corresponding bit in the port output register P1_OUT (see Table 9-3). On a read access, this bit returns 0.
PRx (x = 0-15)	x + 16	w	Port Reset Bit x Setting this bit resets or toggles the corresponding bit in the port output register P1_OUT (see Table 9-3). On a read access, this bit returns 0.

P1_IOCRO

Port 1 Input/Output Control Register 0(40_H)

Reset Value: 2222 2222_H



Field	Bits	Type	Description
PC0B	[3:0]	rw	Alternative Port Control for Port 1 Pin 0 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCRO0B signal.
PC0	[7:4]	rw	Port Control for Port 1 Pin 0 This bit field defines the port functionality according to the coding table (Table 9-13).
PC1B	[11:8]	rw	Alternative Port Control for Port 1 Pin 1 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCRO1B signal.
PC1	[15:12]	rw	Port Control for Port 1 Pin 1 This bit field defines the port functionality according to the coding table (Table 9-13).
PC2B	[19:16]	rw	Alternative Port Control for Port 1 Pin 2 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCRO2B signal.
PC2	[23:20]	rw	Port Control for Port 1 Pin 2 This bit field defines the port functionality according to the coding table (Table 9-13).
PC3B	[27:24]	rw	Alternative Port Control for Port 1 Pin 3 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCRO3B signal.

Parallel Ports (PORTS)

Field	Bits	Type	Description
PC3	[31:28]	rw	Port Control for Port 1 Pin 3 This bit field defines the port functionality according to the coding table (Table 9-13).

Table 9-13 PCx Coding for Port 1

PCx[3:0]	I/O	Output Characteristics	Selected Pull-up/down / Selected Output Function
0000 _B ¹⁾	Direct Input	–	No pull device connected
0001 _B			Not Implemented
0010 _B			Pull-up device connected
0011 _B			Not Implemented
0100 _B	Inverted Input	–	Not Implemented
0101 _B			Not Implemented
0110 _B			Not Implemented
0111 _B			Not Implemented
1000 _B	Output (Direct input)	Push-pull	General purpose Output
1001 _B			Output function ALT1
1010 _B			Output function ALT2
1011 _B			Output function ALT3
1100 _B		Open Drain	Not Implemented
1101 _B			Not Implemented
1110 _B			Not Implemented
1111 _B			Not Implemented

1) This bit field value is default after reset.

Parallel Ports (PORTS)

P1_IOCRA

Port 1 Input/Output Control Register 4(44_H)

Reset Value: 2222 2222_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC7				PC7B				PC6				PC6B			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC5				PC5B				PC4				PC4B			
rw				rw				rw				rw			

Field	Bits	Type	Description
PC4B	[3:0]	rw	Alternative Port Control for Port 1 Pin 4 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCRA4B signal.
PC4	[7:4]	rw	Port Control for Port 1 Pin 4 This bit field defines the port functionality according to the coding table (Table 9-13).
PC5B	[11:8]	rw	Alternative Port Control for Port 1 Pin 5 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCRA5B signal.
PC5	[15:12]	rw	Port Control for Port 1 Pin 5 This bit field defines the port functionality according to the coding table (Table 9-13).
PC6B	[19:16]	rw	Alternative Port Control for Port 1 Pin 6 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCRA6B signal.
PC6	[23:20]	rw	Port Control for Port 1 Pin 6 This bit field defines the port functionality according to the coding table (Table 9-13).
PC7B	[27:24]	rw	Alternative Port Control for Port 1 Pin 7 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCRA7B signal.

Parallel Ports (PORTS)

Field	Bits	Type	Description
PC7	[31:28]	rw	Port Control for Port 1 Pin 7 This bit field defines the port functionality according to the coding table (Table 9-13).

P1_IOC8

Port 1 Input/Output Control Register 8(48_H)

Reset Value: 2222 2222_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC11				PC11B				PC10				PC10B			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC9				PC9B				PC8				PC8B			
rw				rw				rw				rw			

Field	Bits	Type	Description
PC8B	[3:0]	rw	Alternative Port Control for Port 1 Pin 8 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOC8B signal.
PC8	[7:4]	rw	Port Control for Port 1 Pin 8 This bit field defines the port functionality according to the coding table (Table 9-13).
PC9B	[11:8]	rw	Alternative Port Control for Port 1 Pin 9 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOC9B signal.
PC9	[15:12]	rw	Port Control for Port 1 Pin 9 This bit field defines the port functionality according to the coding table (Table 9-13).
PC10B	[19:16]	rw	Alternative Port Control for Port 1 Pin 10 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOC10B signal.

Parallel Ports (PORTS)

Field	Bits	Type	Description
PC10	[23:20]	rw	Port Control for Port 1 Pin 10 This bit field defines the port functionality according to the coding table (Table 9-13).
PC11B	[27:24]	rw	Alternative Port Control for Port 1 Pin 11 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCR11B signal.
PC11	[31:28]	rw	Port Control for Port 1 Pin 11 This bit field defines the port functionality according to the coding table (Table 9-13).

P1_IOC12

Port 1 Input/Output Control Register 12(4C_H)

Reset Value: 2222 2222_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC15				PC15B				PC14				PC14B			
rw				rw				rw				r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC13				PC13B				PC12				PC12B			
rw				rw				rw				rw			

Field	Bits	Type	Description
PC12B	[3:0]	rw	Alternative Port Control for Port 1 Pin 12 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCR12B signal.
PC12	[7:4]	rw	Port Control for Port 1 Pin 12 This bit field defines the port functionality according to the coding table (Table 9-13).
PC13B	[11:8]	rw	Alternative Port Control for Port 1 Pin 13 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCR13B signal.

Parallel Ports (PORTS)

Field	Bits	Type	Description
PC13	[15:12]	rw	Port Control for Port 1 Pin 13 This bit field defines the port functionality according to the coding table (Table 9-13).
PC14B	[19:16]	rw	Alternative Port Control for Port 1 Pin 14 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCR14B signal.
PC14	[23:20]	rw	Port Control for Port 1 Pin 14 This bit field defines the port functionality according to the coding table (Table 9-13).
PC15B	[27:24]	rw	Alternative Port Control for Port 1 Pin 15 This bit field defines the port functionality according to the coding table (Table 9-13) if selected by IO module by setting IOCR15B signal.
PC15	[31:28]	rw	Port Control for Port 1 Pin 15 This bit field defines the port functionality according to the coding table (Table 9-13).

Parallel Ports (PORTS)

P1_IN

Port 1 Input Register

(54_H)

Reset Value: 0000 XXXX_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
Px (x = 0-15)	x	rh	Port Input Bit x This bit indicates the level at the input pin of port P1, pin x. 0 The input level of P1.x is 0. 1 The input level of P1.x is 1.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

Parallel Ports (PORTS)

P1_PDR

Port 1 Pad Driver Mode Register

(58_H)

Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	PD3			0	PD2			0	PD1			0	PD0		
r	rw			r	rw			r	rw			r	rw		

Field	Bits	Type	Description
PD0	[2:0]	rw	Pad Driver Mode for P0.[3:0] Coding see Table 9-1 .
PD0	[6:4]	rw	Pad Driver Mode for P0.[7:4] Coding see Table 9-1 .
PD2	[10:8]	rw	Pad Driver Mode for P0.[11:8] Coding see Table 9-1 .
PD3	[14:12]	rw	Pad Driver Mode for P0.[15:12] Coding see Table 9-1 .
0	3, 7, 11, [31:15]	r	Reserved ; read as 0; should be written with 0.

9.2.3.3 Port 1 Functions

The following table summarizes the signals routed to the Port 1 pads.

Table 9-14 Port 1 Input/Output Functions

Port Pin	I/O ¹⁾	Select	Connected Signal(s)	From / to Module
P1.0	Input		Port Input Register P1_IN.0, D0	Port Input, XMU
	Output	GPIO	Port Output Register P1_OUT.0	Port Output
		ALT1	D0	XMU
		ALT2	TVALIDB	MLI
		ALT3	INT_O0	SCU
P1.1	Input		Port Input Register P1_IN.1, D1	Port Input, XMU
	Output	GPIO	Port Output Register P1_OUT.1	Port Output
		ALT1	D1	XMU
		ALT2	TCLK	MLI
		ALT3	INT_O1	SCU
P1.2	Input		Port Input Register P1_IN.2, SCLK, D2	Port Input, SSC, XMU
	Output	GPIO	Port Output Register P1_OUT.2	Port Output
		ALT1	D2	XMU
		ALT2	TDATA	MLI
		ALT3	INT_O2	SCU
P1.3	Input		Port Input Register P1_IN.3, D3	Port Input, XMU
	Output	GPIO	Port Output Register P1_OUT.3	Port Output
		ALT1	D3	XMU
		ALT2	RREADYB	MLI
		ALT3	RDY	SSC
P1.4	Input		Port Input Register P1_IN.4, RVALIDB, SLS, D4	Port Input, MLI, SSC, XMU
	Output	GPIO	Port Output Register P1_OUT.4	Port Output
		ALT1	D4	XMU
		ALT2	Not implemented	
		ALT3	Not implemented	

Parallel Ports (PORTS)
Table 9-14 Port 1 Input/Output Functions (cont'd)

Port Pin	I/O ¹⁾	Select	Connected Signal(s)	From / to Module
P1.5	Input		Port Input Register P1_IN.5, RCLKB, Half Duplex Control SSC, D5	Port Input, MLI, SSC, XMU
	Output	GPIO	Port Output Register P1_OUT.5	Port Output
		ALT1	D5	XMU
		ALT2	INT_O0	SCU
		ALT3	MRST	SSC
P1.6	Input		Port Input Register P1_IN.6, RDATA B, MTSR, D6	Port Input, MLI, SSC, XMU
	Output	GPIO	Port Output Register P1_OUT.6	Port Output
		ALT1	D6	XMU
		ALT2	Not implemented	
		ALT3	MRST	SSC
P1.7	Input		Port Input Register P1_IN.7, MODE[1], D7	Port Input, SCU, XMU
	Output	GPIO	Port Output Register P1_OUT.7	Port Output
		ALT1	D7	XMU
		ALT2	INT_O1	SCU
		ALT3	INT_O2	SCU
P1.8	Input		Port Input Register P1_IN.8, SR0, D8	Port Input, SCU, XMU
	Output	GPIO	Port Output Register P1_OUT.8	Port Output
		ALT1	D8	XMU
		ALT2	INT_O1	SCU
		ALT3	INT_O2	SCU
P1.9	Input		Port Input Register P1_IN.9, SR1, D9	Port Input, SCU, XMU
	Output	GPIO	Port Output Register P1_OUT.9	Port Output
		ALT1	D9	XMU
		ALT2	INT_O3	SCU
		ALT3	INT_O4	SCU

Parallel Ports (PORTS)
Table 9-14 Port 1 Input/Output Functions (cont'd)

Port Pin	I/O¹⁾	Select	Connected Signal(s)	From / to Module
P1.10	Input		Port Input Register P1_IN.10, SR2, D10	Port Input, SCU, XMU
	Output	GPIO	Port Output Register P1_OUT.10	Port Output
		ALT1	D10	XMU
		ALT2	INT_O0	SCU
		ALT3	INT_O4	SCU
P1.11	Input		Port Input Register P1_IN.11, SR3, D11	Port Input, SCU, XMU
	Output	GPIO	Port Output Register P1_OUT.11	Port Output
		ALT1	D11	XMU
		ALT2	INT_O2	SCU
		ALT3	INT_O3	SCU
P1.12	Input		Port Input Register P1_IN.12, SR7, D12	Port Input, SCU, XMU
	Output	GPIO	Port Output Register P1_OUT.12	Port Output
		ALT1	D12	XMU
		ALT2	INT_O0	SCU
		ALT3	INT_O1	SCU
P1.13	Input		Port Input Register P1_IN.13, SR6, D13	Port Input, SCU, XMU
	Output	GPIO	Port Output Register P1_OUT.13	Port Output
		ALT1	D13	XMU
		ALT2	INT_O2	SCU
		ALT3	INT_O4	SCU
P1.14	Input		Port Input Register P1_IN.14, SR5, D14	Port Input, SCU, XMU
	Output	GPIO	Port Output Register P1_OUT.14	Port Output
		ALT1	D14	XMU
		ALT2	INT_O1	SCU
		ALT3	INT_O4	SCU

Parallel Ports (PORTS)

Table 9-14 Port 1 Input/Output Functions (cont'd)

Port Pin	I/O ¹⁾	Select	Connected Signal(s)	From / to Module
P1.15	Input		Port Input Register P1_IN.15, SR4, D15	Port Input, SCU, XMU
	Output	GPIO	Port Output Register P1_OUT.14	Port Output
		ALT1	D15	XMU
		ALT2	INT_O1	SCU
		ALT3	INT_O3	SCU

1) For this port, all pins can be read as GPIO, from the Port Input Register.

9.2.4 Port 2 Registers

Port 2 is a general purpose 16-bit unidirectional input port.

9.2.4.1 Overview

The port registers of Port 2 are shown in [Figure 9-7](#).

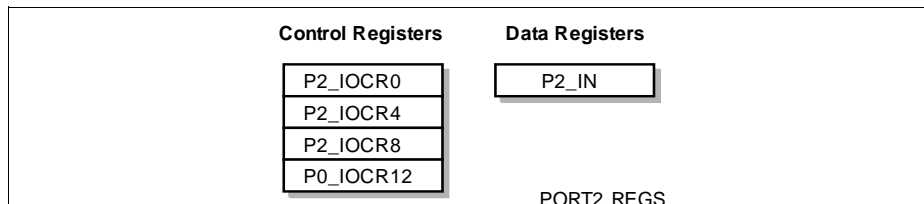


Figure 9-7 Port 2 Register Overview

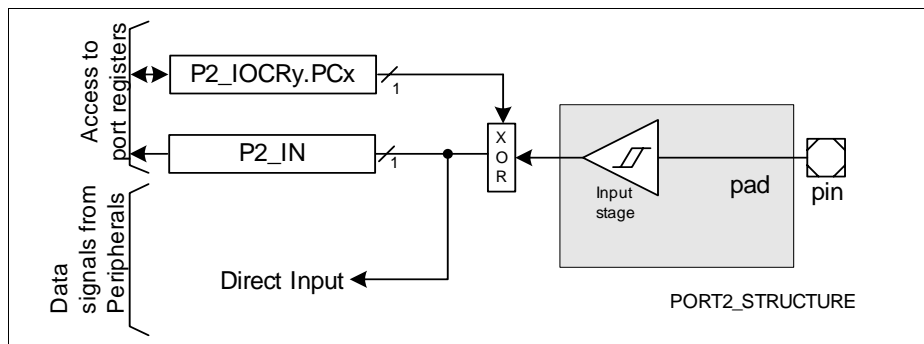


Figure 9-8 Port 2 Structure

9.2.4.2 Port 2 Register Implementations

Table 9-15 Registers Address SpacePort_2 Kernel Register Address Space

Module	Base Address	End Address	Note
P2	0000 0A00 _H	0000 0AFF _H	256 byte P0-P2 have Common Address Space

Table 9-16 Port 2 Registers

Register Short Name	Register Long Name	Relative Address ¹⁾	Description see
	Reserved	60 _H	
	Reserved	64 _H	
	Reserved	68 _H	
	Reserved	6C _H	
P2_IOCRO	Port 2 Input/Output Control Register 0	70 _H	Page 9-47
P2_IOCRR4	Port 2 Input/Output Control Register 4	74 _H	Page 9-49
P2_IOCRR8	Port 2 Input/Output Control Register 8	78 _H	Page 9-50
P2_IOCRR12	Port 2 Input/Output Control Register 12	7C _H	Page 9-51
	Reserved	80 _H	
P2_IN	Port 2 Input Register	84 _H	Page 9-52
	Reserved	88 _H	
	Reserved	8C _H	

1) The relative address indicates the offset to the base address.

Parallel Ports (PORTS)

P2_IOCR0

Port 2 Input/Output Control Register 0(70_H)

Reset Value: 2020 2020_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC3				0				PC2				0			
rw				r				rw				r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC1				0				PC0				0			
rw				r				rw				r			

Field	Bits	Type	Description
PC0	[7:4]	rw	Port Control for Port 2 Pin 0 This bit field defines the port functionality according to the coding table (Table 9-17).
PC1	[15:12]	rw	Port Control for Port 2 Pin 1 This bit field defines the port functionality according to the coding table (Table 9-17).
PC2	[23:20]	rw	Port Control for Port 2 Pin 2 This bit field defines the port functionality according to the coding table (Table 9-17).
PC3	[31:28]	rw	Port Control for Port 2 Pin 3 This bit field defines the port functionality according to the coding table (Table 9-17).
0	[3:0], [11:8], [19:16], [27:24]	r	Reserved ; read as 0; should be written with 0.

Table 9-17 PC13 Coding for Port 2

PCx[3:0]	I/O	Output Characteristics	Selected Pull-up/down / Selected Output Function
0000 _B ¹⁾	Direct Input	—	No pull device connected
0001 _B			Not Implemented
0010 _B			Pull-up device connected
0011 _B			Not Implemented
0100 _B	Inverted Input	—	Not Implemented
0101 _B			Not Implemented
0110 _B			Not Implemented
0111 _B			Not Implemented
1000 _B	Output (Direct input)	Push-pull	Not Implemented
1001 _B			Not Implemented
1010 _B			Not Implemented
1011 _B			Not Implemented
1100 _B		Open Drain	Not Implemented
1101 _B			Not Implemented
1110 _B			Not Implemented
1111 _B			Not Implemented

1) This bit field value is default after reset.

P2_IOC4

Port 2 Input/Output Control Register 4(74_H)

Reset Value: 2020 2020_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC7				0				PC6				0			
rw				r				rw				r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC5				0				PC4				0			
rw				r				rw				r			

Field	Bits	Type	Description
PC4	[7:4]	rw	Port Control for Port 2 Pin 4 This bit field defines the port functionality according to the coding table (Table 9-17).
PC5	[15:12]	rw	Port Control for Port 2 Pin 5 This bit field defines the port functionality according to the coding table (Table 9-17).
PC6	[23:20]	rw	Port Control for Port 2 Pin 6 This bit field defines the port functionality according to the coding table (Table 9-17).
PC7	[31:28]	rw	Port Control for Port 2 Pin 7 This bit field defines the port functionality according to the coding table (Table 9-17).
0	[3:0], [11:8], [19:16], [27:24]	r	Reserved ; read as 0; should be written with 0.

P2_IOC8

Port 2 Input/Output Control Register 8(78_H)

Reset Value: 2020 2020_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC11				0				PC10				0			
rw				r				rw				r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC9				0				PC8				0			
rw				r				rw				r			

Field	Bits	Type	Description
PC8	[7:4]	rw	Port Control for Port 2 Pin 8 This bit field defines the port functionality according to the coding table (Table 9-17).
PC9	[15:12]	rw	Port Control for Port 2 Pin 9 This bit field defines the port functionality according to the coding table (Table 9-17).
PC10	[23:20]	rw	Port Control for Port 2 Pin 10 This bit field defines the port functionality according to the coding table (Table 9-17).
PC11	[31:28]	rw	Port Control for Port 2 Pin 11 This bit field defines the port functionality according to the coding table (Table 9-17).
0	[3:0], [11:8], [19:16], [27:24]	r	Reserved ; read as 0; should be written with 0.

P2_IOC12

Port 2 Input/Output Control Register 12(7C_H)

Reset Value: 2020 2020_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC15				0				PC14				0			
rw				r				rw				r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC13				0				PC12				0			
rw				r				rw				r			

Field	Bits	Type	Description
PC12	[7:4]	rw	Port Control for Port 2 Pin 12 This bit field defines the port functionality according to the coding table (Table 9-17).
PC13	[15:12]	rw	Port Control for Port 2 Pin 13 This bit field defines the port functionality according to the coding table (Table 9-17).
PC14	[23:20]	rw	Port Control for Port 2 Pin 14 This bit field defines the port functionality according to the coding table (Table 9-17).
PC15	[31:28]	rw	Port Control for Port 2 Pin 15 This bit field defines the port functionality according to the coding table (Table 9-17).
0	[3:0], [11:8], [19:16], [27:24]	r	Reserved ; read as 0; should be written with 0.

Parallel Ports (PORTS)

P2_IN

Port 2 Input Register

(84_H)

Reset Value: 0000 XXXX_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
Px (x = 0-15)	x	rh	Port Input Bit x This bit indicates the level at the input pin of port P2, pin x. 0 The input level of P2.x is 0. 1 The input level of P2.x is 1.
0	[31:16]	r	Reserved ; read as 0; should be written with 0.

9.2.4.3 Port 2 Functions

The following table summarizes the signals routed to the Port 2 pads.

Table 9-18 Port 2 Input/Output Functions

Port Pin	I/O ¹⁾	Select	Connected Signal(s)	From / to Module
P2.0	Input		Port Input Register P2_IN.0, A0,BC[0]	Port Input, SCU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.1	Input		Port Input Register P2_IN.1, A1	Port Input, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.2	Input		Port Input Register P2_IN.2, A2	Port Input, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.3	Input		Port Input Register P2_IN.3, A3	Port Input, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.4	Input		Port Input Register P2_IN.4, A4	Port Input, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	

Parallel Ports (PORTS)
Table 9-18 Port 2 Input/Output Functions (cont'd)

Port Pin	I/O¹⁾	Select	Connected Signal(s)	From / to Module
P2.5	Input		Port Input Register P2_IN.5, A5	Port Input, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.6	Input		Port Input Register P2_IN.6, A6	Port Input, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.7	Input		Port Input Register P2_IN.7, A7	Port Input, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.8	Input		Port Input Register P2_IN.8, A8	Port Input, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.9	Input		Port Input Register P2_IN.9, A9	Port Input, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.10	Input		Port Input Register P2_IN.10, A10	Port Input, SCU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	

Parallel Ports (PORTS)

Table 9-18 Port 2 Input/Output Functions (cont'd)

Port Pin	I/O¹⁾	Select	Connected Signal(s)	From / to Module
P2.11	Input		Port Input Register P2_IN.11, A11	Port Input, SCU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.12	Input		Port Input Register P2_IN.12, $\overline{\text{BYPASS}}$, $\overline{\text{CSFPI}}$	Port Input, SCU, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.13	Input		Port Input Register P2_IN.13, $\overline{\text{WR}}$	Port Input, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.14	Input		Port Input Register P2_IN.14, TREADYB, BC[1]	Port Input, MLI, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	
P2.15	Input		Port Input Register P2_IN.15, SR15, $\overline{\text{RD}}$	Port Input, SCU, XMU
	Output	GPIO	Not Implemented	
		ALT1	Not Implemented	
		ALT2	Not Implemented	
		ALT3	Not Implemented	

1) For this port, all pins can be read as GPIO, from the Port Input Register.

9.3 Address Map

In the SAK-CIC310-OSMX2HT, the registers of the Ports module are located in the following address Range:

- Module Base Address = 0000 0A00_H
Module End Address = 0000 0AFF_H
- Absolute Register Address = Module Base Address + Offset Address (offset addresses see [Table 9-11](#), [Table 9-12](#), and [Table 9-16](#))

10 Initialization of the SAK-CIC310-OSMX2HT Chip

The initialization of the SAK-CIC310-OSMX2HT is triggered by the $\overline{\text{PORST}}$ pin activation. The initialization procedure consists of 3 parts:

- The reset
- Initialization by an Initialization Move Engine (IME)
- Software (application) initialization by using one of the activated host interface (serial or parallel)

The reset signal switches every register into its defined reset state, latches the MODE pins, and JTAG Enable Pin, and triggers the Initialization Move Engine (IME) depending on the latched value.

The second part of the initialization is performed thereafter by the Initialization Move Engine (IME), which executes a predefined number of write transactions from a dedicated internal memory, which contains the address and the data to be written for all required write transactions. There are 3 write transaction sets, one for the initialization for MLI communication, one for the initialization for XMU communication, and one for the initialization for SSC/SPI communication. The Initialization Move Engine (IME) selects the correct set depending on the latched MODE (MODEL) value.

If the reset value of a register equals the required initialization value, then the re-initialization by the Initialization Move Engine (IME) is not necessary. The required initialization values are listed in [Table 10-2](#) and [Table](#) for the MLI, [Table 10-13](#), and [Table 10-17](#) for the SSC communication.

There are three possibilities for the communication depending on the latched MODE pins: MLI, XMU, or SSC. The coverage of the Initialization Move Engine (IME) should only be, what is necessary for a first communication via one of the serial or parallel interfaces.

The initialization by the Initialization Move Engine (IME) is done via the crossbar switch.

Initialization of the SAK-CIC310-OSMX2HT Chip

10.1 Initialization for MLI Communication

Using the MLI the SAK-CIC310-OSMX2HT must be able to receive data from the external host transmitter after the initialization by the Initialization Move Engine (IME). Further steps may then be taken by the external host via application software. The startup procedure ([Chapter 7.2.4](#)) could be done using the already configured read communication. Furthermore the PLL Unlock Interrupt has to be driven to be able to detect PLL faults during start-up. Therefore the port_0 has to be set-up.

The registers of the port 0, port_1, and the MLI need to be initialized by the Initialization Move Engine (IME).

10.1.1 The Initialization Values of Port 0

The pin 8 of Port 0 communicating PLL Lock faults, being a push-pull output of ALT2 signal. To reduce EMI all other ports are switched to medium driver strength. Two register of port 0 are programmed by the Initialization Move Engine (IME) respective to [Table 10-1](#).

Table 10-1 Port 0 Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
P0_IOCRR8	Port 0 Input/Output Control Register 8	0000 0A18 _H	2020 20A0 _H
P0_IOCRR12	Port 0 Input/Output Control Register 12	0000 0A1C _H	0000 2220 _H
P0_PDR	Port 0 Pad Driver Mode Register	0000 0A28 _H	0000 4400 _H

10.1.2 The Initialization Values of Port 1

The pins 0 to 3 of Port 1 being MLI output signals have to be switched to push-pull outputs of the ALT2 signals. To reduce EMI all other ports are switched to medium driver strength. Two registers of port 0 are programmed by the Initialization Move Engine (IME) in respective to [Table 10-2](#).

Table 10-2 Port 1 Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
P1_IOCRR0	Port 1 Input/Output Control Register 0	0000 0A40 _H	A0A0 A0A0 _H
P1_IOCRR4	Port 1 Input/Output Control Register 4	0000 0A44 _H	2000 0000 _H
P1_PDR	Port 1 Pad Driver Mode Register	0000 0A58 _H	0000 4440 _H

Initialization of the SAK-CIC310-OSMX2HT Chip

10.1.3 The Initialization Values of Port 2

The pin 14 of Port 2 being MLI input signal has to have a disabled pull-up device. All other pins are set as input with enabled pull-up device by the Initialization Move Engine (IME) in respective to [Table 10-3](#).

Table 10-3 Port 2 Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
P2_IOC4	Port 2 Input/Output Control Register 4	0000 0A74 _H	2020 2020 _H
P2_IOC8	Port 2 Input/Output Control Register 8	0000 0A78 _H	2020 2020 _H
P2_IOC12	Port 2 Input/Output Control Register 12	0000 0A7C _H	2000 2020 _H

10.1.4 Initialization Values of MLI

After reset the receiver is still in reset mode. Then the Initialization Move Engine (IME) has to update the MLI_IOC, MLI_SCR, MLI_RCR, and MLI_TCR according to [Table 10-4](#).

During Reset the MLI signals are switched to setting A, by programming OICR setting B is initialized.

Table 10-4 MLI Kernel Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
MLI_IOC	Output Input Control Register	0000 02B4 _H	3241 9902 _H
MLI_SCR	Set Clear Register	0000 0294 _H	0000 0010 _H
MLI_RCR	Receiver Control Register	0000 0268 _H	0000 0000 _H
MLI_TCR	Transmitter Control Register	0000 0210 _H	0000 1341 _H

10.1.5 The Initialization Values of the Service Request Control

The SAK-CIC310-OSMX2HT uses the service request pin SR8 to communicate to the host a PLL_loss_lock event after initialisation by the user. The internal generated interrupt signal has to be respectively routed to INTO1. Therefore the SRCR.INSEL1 bit field has to be set to select the PLL_loss_lock signal for the respective interrupt output signal (INT01) and the alternate Output of the port 0 pin 0 set to A0_H. The SCU_SRCR register is programmed by the Initialization Move Engine (IME) respective to [Table 10-18](#).

Initialization of the SAK-CIC310-OSMX2HT Chip

Table 10-5 SCU Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
SCU_SRCR	Service Request Control Register	0000 0858 _H	0000 0070 _H

10.2 Initialization for XMU Communication

The registers of the port 0, port 1, and the XMU need to be initialized by the Initialization Move Engine (IME).

10.2.1 The Initialization Values of Port 0

The pin 13 of Port 0 communicating PLL Lock faults, being a push-pull output on ALT2. To reduce EMI the other port nibbles (4 bit group) are either switched to medium driver strength or strong driver strength, medium edge. Two register of port 0 are programmed by the Initialization Move Engine (IME) respective to [Table 10-1](#).

Table 10-6 Port 0 Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
P0_IOC8	Port 0 Input/Output Control Register 8	0000 0A18 _H	2020 2000 _H
P0_IOC12	Port 0 Input/Output Control Register 12	0000 0A1C _H	0000 0990 _H
P0_PDR	Port 0 Pad Driver Mode Register	0000 0A28 _H	0000 1400 _H

10.2.2 The Initialization Values of Port 1

The pins 0 to 7 of Port 1 being operation dependant input and output signals have to be switched to input and push-pull outputs of the ALT1 signals. To reduce EMI all other ports are switched to medium driver strength. Three registers of port 0 are programmed by the Initialization Move Engine (IME) in respective to [Table 10-7](#). This programming enables 8-bit and 16-bit external to internal write operations (host write) and only 8-bit external to internal read operations (host read). Three register of port 1 are programmed by the Initialization Move Engine (IME) respective to [Table 10-7](#).

Initialization of the SAK-CIC310-OSMX2HT Chip

Table 10-7 Port 1 Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
P1_IOCRO	Port 1 Input/Output Control Register 0	0000 0A40 _H	0909 0909 _H
P1_IOCRO4	Port 1 Input/Output Control Register 4	0000 0A44 _H	0909 0909 _H
P1_IOCRO8	Port 1 Input/Output Control Register 8	0000 0A48 _H	0000 0000 _H
P1_IOCRO12	Port 1 Input/Output Control Register 12	0000 0A4C _H	0000 0000 _H
P1_PDR	Port 1 Pad Driver Mode Register	0000 0A58 _H	0000 4400 _H

10.2.2.1 XMU with 16-bit Data Width

If using the XMU with a data width of 16-bit (D0...D15), additional port registers have to be programmed via XMU external to internal write operations (host write, in general 16 bit wide), else wise only 8-bit data width (D0...D7) are enabled for external to internal read operations (host reads). The application software has to write new values to three register of port 1 as listed in [Table 10-8](#) to enable read functionality of 16-bit data width communication.

Table 10-8 Port1 Registers to be Initialized by Host for 16-Bit Data Width

Register Short Name	Register Long Name	Address	Data to be written by Application
P1_IOCRO8	Port 1 Input/Output Control Register 8	0000 0A48 _H	0909 0909 _H
P1_IOCRO12	Port 1 Input/Output Control Register 12	0000 0A4C _H	0909 0909 _H
P1_PDR	Port 1 Pad Driver Mode Register	0000 0A58 _H	0000 0000 _H

10.2.3 The Initialization Values of Port 2

The pins of Port 2 are in general inputs of the XMU (exception pin 12) and the pull-up devices have to be disabled by the Initialization Move Engine (IME) in respective to [Table 10-9](#) to decrease load for the host controller.

Initialization of the SAK-CIC310-OSMX2HT Chip

Table 10-9 Port 2 Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
P2_IOCRO	Port 2 Input/Output Control Register 0	0000 0A70 _H	0000 0000 _H
P2_IOCRO4	Port 2 Input/Output Control Register 4	0000 0A74 _H	0000 0000 _H
P2_IOCRO8	Port 2 Input/Output Control Register 8	0000 0A78 _H	0000 0000 _H
P2_IOCRO12	Port 2 Input/Output Control Register 12	0000 0A7C _H	0000 0020 _H

10.2.4 Initialization Values of the XMU

After reset the XMU is disabled. The Initialization Move Engine (IME) has to update the XMU_CON according to [Table 10-10](#).

Table 10-10 XMU Kernel Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
XMU_CON	XMU Global Control Register	0000 0864 _H	0000 0002 _H

10.2.5 The Initialization Values of the Service Request Control

The SAK-CIC310-OSMX2HT uses the service request pin SR9 to communicate to the host a PLL_loss_lock event after initialization by the user. The internal generated interrupt signal has to be respectively routed to INTO0. Therefore the SRCR.INSEL0 bit field has to be set to select the PLL_loss_lock signal for the respective interrupt output signal (INTO0). The STCU_SRCR register is programmed by the Initialization Move Engine (IME) respective to [Table 10-18](#).

Table 10-11 SCU Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
SCU_SRCR	Service Request Control Register	0000 0858 _H	0000 0007 _H

10.3 Initialization for SSC/SPI Communication

Using the SSC of the SAK-CIC310-OSMX2HT must be able to receive data from the external host transmitter after the initialization by the Initialization Move Engine (IME). Further steps may then be taken by the external host via application software to

Initialization of the SAK-CIC310-OSMX2HT Chip

configure the SSC to either half-duplex or full-duplex mode. Furthermore the PLL Unlock Interrupt has to be driven to be able to detect PLL faults during start-up.

Registers of the port 1, port 0, and the SSC need to be initialized by the Initialization Move Engine (IME).

10.3.1 The Initialization Values of Port 0

The pin 8 of Port 0 communicating PLL Lock faults, being a push-pull output of ALT2 signal. To reduce EMI all other ports are switched to medium driver strength. Two register of port 0 are programmed by the Initialization Move Engine (IME) respective to [Table 10-12](#).

Table 10-12 Port 0 Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
P0_IOC8	Port 0 Input/Output Control Register 8	0000 0A18 _H	2020 20A0 _H
P0_IOC12	Port 0 Input/Output Control Register 12	0000 0A1C _H	0000 2220 _H
P0_PDR	Port 0 Pad Driver Mode Register	0000 0A28 _H	0000 4400 _H

10.3.2 The Initialization Values of Port 1

The port 1 pins being outputs of the SSC (MRST) depend on the mode the SSC is used in half-duplex or full duplex mode. Therefore this has to be configured by host write operations. Only the port 1 pin 3 being the SSC ready signal (RDY) is configured as push-pull outputs of ALT3 signal. To reduce EMI all other ports are switched to medium driver strength. Two register of port 1 are programmed by the Initialization Move Engine (IME) respective to [Table 10-13](#).

Table 10-13 Port 1 Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
P1_IOC0	Port 1 Input/Output Control Register 0	0000 0A40 _H	B000 2020 _H
P1_IOC4	Port 1 Input/Output Control Register 4	0000 0A44 _H	2000 2020 _H
P1_PDR	Port 1 Pad Driver Mode Register	0000 0A58 _H	0000 4400 _H

10.3.2.1 SSC Full-Duplex Mode

If using the SSC in full duplex mode (separate MRST and MTSR lines), one additional port register has to be programmed via SSC external to internal write operation (host

Initialization of the SAK-CIC310-OSMX2HT Chip

write), else wise no external to internal read operations (host reads) is supported. The application software has to write a new value to one register of port 1 as listed in [Table 10-14](#) to enable the bidirectional full-duplex communication of the SSC.

Table 10-14 Port 1 Registers to be Initialized by Host for Full-Duplex Communication

Register Short Name	Register Long Name	Address	Data to be written by Application
P1_IOC4	Port 1 Input/Output Control Register 4	0000 0A44 _H	2000 B000 _H

10.3.2.2 SSC Half-Duplex Mode

If using the SSC in half-duplex mode (common MRST and MTSR line), two additional registers, a port and a general control register, have to be programmed via SSC external to internal write operations (host write), else wise no external to internal read operations (host reads) is supported. The application software has to write new values to the two register as listed in [Table 10-15](#) to enable the bidirectional full-duplex communication of the SSC.

Table 10-15 Port 1 Registers to be initialized by Host for Half-Duplex Communication

Register Short Name	Register Long Name	Address	Data to be written by Application
P1_IOC4	Port 1 Input/Output Control Register 4	0000 0A44 _H	000B 0000 _H
SSC_ERRCUM	SSC Cumulative Error Register	0000 085C _H	0000 1000 _H

10.3.3 The Initialization Values of Port 2

All pins are set as input with enabled pull-up device by the Initialization Move Engine (IME) in respective to [Table 10-16](#).

Initialization of the SAK-CIC310-OSMX2HT Chip

Table 10-16 Port 2 Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
P2_IOCR4	Port 2 Input/Output Control Register 4	0000 0A74 _H	2020 2020 _H
P2_IOCR8	Port 2 Input/Output Control Register 8	0000 0A78 _H	2020 2020 _H
P2_IOCR12	Port 2 Input/Output Control Register 12	0000 0A7C _H	2020 2020 _H

10.3.4 The Initialization Values of the SSC

During Reset the SSC signals are switched to setting A (Receive input line MRSTA, Receive input line MTSRA, Slave mode clock input line SCLKA). The Initialization Move Engine (IME) then selects setting B by programming the PISEL register.

The SSC is internally configured via the control register SSC_CON:

- Data width SSC_CON.BM of 16,
- A heading SSC_CON.HB of MSB first,
- Phase control SSC_CON.PH with 0 (Shift transmit data on the leading clock edge, latch on trailing edge),
- A clock polarity SSC_CON.PO with 1 (Idle clock line is high, the leading clock edge is high-to-low transition).
- Interrupts enabled.
- The SSC acts as slave
- The SSC is enabled.

The reset value of the baud rate Timer Reload Register (SSC_BR) is 0000 0063_H generating a clock signal of 200 kHz at the SCLK pin if the SAK-CIC310-OSMX2HT device is driven by a 40 MHz clock source.

Table 10-17 SSC Kernel Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
SSC_PISEL	Port Input Select Register	0000 0904 _H	0000 000F _H
SSC_BR	Baud Rate Timer Reload Register	0000 0914 _H	0000 0063 _H
SSC_CON	Control Register	0000 0910 _H	0000 075F _H
SSC_CON	Control Register	0000 0910 _H	0000 875F _H

Note: If using the SSC half-duplex mode, the application software has to set an additional SSC control register as listed in [Table 10-15](#).

Initialization of the SAK-CIC310-OSMX2HT Chip

10.3.5 The Initialization Values of the Service Request Control

The SAK-CIC310-OSMX2HT uses the service request pin SR8 to communicate to the host a PLL_loss_lock event after initialization by the user. The internal generated interrupt signal has to be respectively routed to INT01 as outputs. Therefore the SRCR.INSEL1 bit field has to be set to select the PLL_loss_lock signal for the respective interrupt output signal (INT01). The SCU_SRCR register is programmed by the Initialization Move Engine (IME) respective to [Table 10-18](#).

Table 10-18 Port 1 Registers Initialized by Initialization Move Engine (IME)

Register Short Name	Register Long Name	Address	Data Written by IME
SCU_SRCR	Service Request Control Register	0000 0858 _H	0000 0070 _H

11 Register Description

This chapter defines all registers of the SAK-CIC310-OSMX2HT. It also defines the read/write access rights of the specific address ranges and registers. The summary of the SAK-CIC310-OSMX2HT registers is split into chapters in accordance to the implemented modules.

11.1 Overview

Throughout the tables in this chapter, the “Access Mode” “Read” and “Write”, and “Reset Values” columns indicate access rights and values using symbols listed in [Table 11-1](#).

Table 11-1 Address Map Symbols

Symbol	Description
U	Access Mode: access permitted in User Mode 0 or 1 (not supported for SAK-CIC310-OSMX2HT). Reset Value: value or bit is not changed by a reset operation.
SV	Access permitted in Supervisor Mode (only mode of SAK-CIC310-OSMX2HT).
R	Read-only register.
32	Only 32-bit word accesses are permitted to that register/address range.
E	Endinit protected register/address.
NC	No change, indicated register is not changed on a write operation.
BE	Indicates that an access to this address range generates a Bus Error (ERRSR.BER0, ERRSR.BER0). Write Operations are ignored. Read Operations deliver a 0.
nBE	Indicates that no Bus Error is generated when accessing this address range, even though it is either an access to an undefined address or the access does not follow the given rules.
X	Product Specific value or bit.
U	Undefined value or bit.

Note: All accesses of the SAK-CIC310-OSMX2HT, regardless of the host interface in use, are executed in Supervisor Mode.

Register Description
11.1.1 FlexRay Communication Controller (E-Ray) Registers

Table 11-2 shows all registers associated with the FlexRay Communication Controller (E-Ray) Kernel.

Table 11-2 Address Map of FlexRay Communication Controller (E-Ray)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
FlexRay Communication Controller (E-Ray)					
Customer Registers					
ERAY_CUST0	Product Identification Register	0000 1000 _H	SV,U	nBE	C100 0100 _H
ERAY_CUST1	Busy Control Register	0000 1004 _H	SV,U	SV,U	0000 0000 _H
ERAY_ID	Module Identification Register	0000 1008 _H	SV,U	SV,U	0044 C0xx _H
ERAY_CUST3	Time-out Counter Register	0000 100C _H	SV,U	SV,U	0000 0000 _H
Special Registers					
ERAY_TEST1	Test Register 1	0000 1010 _H	SV,U	SV,U	0000 0300 _H
ERAY_TEST2	Test Register 2	0000 1014 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 1018 _H	nBE	nBE	0000 0000 _H
ERAY_LCK	Lock Register	0000 101C _H	SV,U	SV,U	0000 0000 _H
Interrupt Registers					
ERAY_EIR	Error Interrupt Register	0000 1020 _H	SV,U	SV,U	0000 0000 _H
ERAY_SIR	Status Interrupt Register	0000 1024 _H	SV,U	SV,U	0000 0000 _H
ERAY_EILS	Error Interrupt Line Select	0000 1028 _H	SV,U	SV,U	0000 0000 _H
ERAY_SILS	Status Interrupt Line Select	0000 102C _H	SV,U	SV,U	0303 FFFF _H
ERAY_EIES	Error Interrupt Enable Set	0000 1030 _H	SV,U	SV,U	0000 0000 _H
ERAY_EIER	Error Interrupt Enable Reset	0000 1034 _H	SV,U	SV,U	0000 0000 _H
ERAY_SIES	Status Interrupt Enable Set	0000 1038 _H	SV,U	SV,U	0000 0000 _H

Register Description
Table 11-2 Address Map of FlexRay Communication Controller (E-Ray) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
FlexRay Communication Controller (E-Ray)					
ERAY_SIER	Status Interrupt Enable Reset	0000 103C _H	SV,U	SV,U	0000 0000 _H
ERAY_ILE	Interrupt Line Enable	0000 1040 _H	SV,U	SV,U	0000 0000 _H
ERAY_T0C	Timer 0 Configuration	0000 1044 _H	SV,U	SV,U	0000 0000 _H
ERAY_T1C	Timer 1 Configuration	0000 1048 _H	SV,U	SV,U	0002 0000 _H
ERAY_STPW1	Stop Watch Register 1	0000 104C _H	SV,U	SV,U	0000 0000 _H
ERAY_STPW2	Stop Watch Register 2	0000 1050 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 1054 _H - 0000 107C _H	nBE	nBE	0000 0000 _H
Communication Controller Control Registers					
ERAY_SUCC1	SUC Configuration Register 1	0000 1080 _H	SV,U	SV,U	0C40 1080 _H
ERAY_SUCC2	SUC Configuration Register 2	0000 1084 _H	SV,U	SV,U	0100 0504 _H
ERAY_SUCC3	SUC Configuration Register 3	0000 1088 _H	SV,U	SV,U	0000 0011 _H
ERAY_NEMC	NEM Configuration Register	0000 108C _H	SV,U	SV,U	0000 0000 _H
ERAY_PRTC1	PRT Configuration Register 1	0000 1090 _H	SV,U	SV,U	084C 0633 _H
ERAY_PRTC2	PRT Configuration Register 2	0000 1094 _H	SV,U	SV,U	0F2D 0A0E _H
ERAY_MHDC	MHD Configuration Register	0000 1098 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 109C _H	nBE	nBE	0000 0000 _H
ERAY_GTUC01	GTU Configuration Register 1	0000 10A0 _H	SV,U	SV,U	0000 0280 _H
ERAY_GTUC02	GTU Configuration Register 2	0000 10A4 _H	SV,U	SV,U	0002 000A _H
ERAY_GTUC03	GTU Configuration Register 3	0000 10A8 _H	SV,U	SV,U	0202 0000 _H

Register Description
Table 11-2 Address Map of FlexRay Communication Controller (E-Ray) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
FlexRay Communication Controller (E-Ray)					
ERAY_GTUC04	GTU Configuration Register 4	0000 10AC _H	SV,U	SV,U	0008 0007 _H
ERAY_GTUC05	GTU Configuration Register 5	0000 10B0 _H	SV,U	SV,U	0E00 0000 _H
ERAY_GTUC06	GTU Configuration Register 6	0000 10B4 _H	SV,U	SV,U	0002 0000 _H
ERAY_GTUC07	GTU Configuration Register 7	0000 10B8 _H	SV,U	SV,U	0002 0004 _H
ERAY_GTUC08	GTU Configuration Register 8	0000 10BC _H	SV,U	SV,U	0000 0002 _H
ERAY_GTUC09	GTU Configuration Register 9	0000 10C0 _H	SV,U	SV,U	0000 0101 _H
ERAY_GTUC10	GTU Configuration Register 10	0000 10C4 _H	SV,U	SV,U	0002 0005 _H
ERAY_GTUC11	GTU Configuration Register 11	0000 10C8 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 10CC _H	nBE	nBE	0000 0000 _H
–	Reserved	0000 10D0 _H - 0000 10FC _H	nBE	nBE	0000 0000 _H
Communication Controller Status Registers					
ERAY_CCSV	CC Status Vector	0000 1100 _H	SV,U	nBE	0010 4000 _H
ERAY_CCEV	CC Error Vector	0000 1104 _H	SV,U	nBE	0000 0000 _H
–	Reserved	0000 1108 _H	nBE	nBE	0000 0000 _H
–	Reserved	0000 110C _H	nBE	nBE	0000 0000 _H
ERAY_SCV	Slot Counter Value	0000 1110 _H	SV,U	nBE	0000 0000 _H
ERAY_MTCCV	Macrotick and Cycle Counter Value	0000 1114 _H	SV,U	nBE	0000 0000 _H
ERAY_RCV	Rate Correction Value	0000 1118 _H	SV,U	nBE	0000 0000 _H
ERAY_OCV	Offset Correction Value	0000 111C _H	SV,U	nBE	0000 0000 _H
ERAY_SFS	Sync Frame Status	0000 1120 _H	SV,U	nBE	0000 0000 _H

Register Description
Table 11-2 Address Map of FlexRay Communication Controller (E-Ray) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
FlexRay Communication Controller (E-Ray)					
ERAY_SWNIT	Symbol Window and NIT Status	0000 1124 _H	SV,U	nBE	0000 0000 _H
ERAY_ACS	Aggregated Channel Status	0000 1128 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 112C _H	SV,U	nBE	0000 0000 _H
ERAY_ESID01	Even Sync ID 1	0000 1130 _H	SV,U	nBE	0000 0000 _H
ERAY_ESID02	Even Sync ID 2	0000 1134 _H	SV,U	nBE	0000 0000 _H
ERAY_ESID03	Even Sync ID 3	0000 1138 _H	SV,U	nBE	0000 0000 _H
ERAY_ESID04	Even Sync ID 4	0000 113C _H	SV,U	nBE	0000 0000 _H
ERAY_ESID05	Even Sync ID 5	0000 1140 _H	SV,U	nBE	0000 0000 _H
ERAY_ESID06	Even Sync ID 6	0000 1144 _H	SV,U	nBE	0000 0000 _H
ERAY_ESID07	Even Sync ID 7	0000 1148 _H	SV,U	nBE	0000 0000 _H
ERAY_ESID08	Even Sync ID 8	0000 114C _H	SV,U	nBE	0000 0000 _H
ERAY_ESID09	Even Sync ID 9	0000 1150 _H	SV,U	nBE	0000 0000 _H
ERAY_ESID10	Even Sync ID 10	0000 1154 _H	SV,U	nBE	0000 0000 _H
ERAY_ESID11	Even Sync ID 11	0000 1158 _H	SV,U	nBE	0000 0000 _H
ERAY_ESID12	Even Sync ID 12	0000 115C _H	SV,U	nBE	0000 0000 _H
ERAY_ESID13	Even Sync ID 13	0000 1160 _H	SV,U	nBE	0000 0000 _H
ERAY_ESID14	Even Sync ID 14	0000 1164 _H	SV,U	nBE	0000 0000 _H
ERAY_ESID15	Even Sync ID 15	0000 1168 _H	SV,U	nBE	0000 0000 _H
–	Reserved	0000 116C _H	SV,U	nBE	0000 0000 _H
ERAY_OSID01	Odd Sync ID 1	0000 1170 _H	SV,U	nBE	0000 0000 _H
ERAY_OSID02	Odd Sync ID 2	0000 1174 _H	SV,U	nBE	0000 0000 _H
ERAY_OSID03	Odd Sync ID 3	0000 1178 _H	SV,U	nBE	0000 0000 _H
ERAY_OSID04	Odd Sync ID 4	0000 117C _H	SV,U	nBE	0000 0000 _H
ERAY_OSID05	Odd Sync ID 5	0000 1180 _H	SV,U	nBE	0000 0000 _H
ERAY_OSID06	Odd Sync ID 6	0000 1184 _H	SV,U	nBE	0000 0000 _H
ERAY_OSID07	Odd Sync ID 7	0000 1188 _H	SV,U	nBE	0000 0000 _H

Register Description
Table 11-2 Address Map of FlexRay Communication Controller (E-Ray) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	

FlexRay Communication Controller (E-Ray)

ERAY_OSID08	Odd Sync ID 8	0000 118C _H	SV,U	nBE	0000 0000 _H
ERAY_OSID09	Odd Sync ID 9	0000 1190 _H	SV,U	nBE	0000 0000 _H
ERAY_OSID10	Odd Sync ID 10	0000 1194 _H	SV,U	nBE	0000 0000 _H
ERAY_OSID11	Odd Sync ID 11	0000 1198 _H	SV,U	nBE	0000 0000 _H
ERAY_OSID12	Odd Sync ID 12	0000 119C _H	SV,U	nBE	0000 0000 _H
ERAY_OSID13	Odd Sync ID 13	0000 11A0 _H	SV,U	nBE	0000 0000 _H
ERAY_OSID14	Odd Sync ID 14	0000 11A4 _H	SV,U	nBE	0000 0000 _H
ERAY_OSID15	Odd Sync ID 15	0000 11A8 _H	SV,U	nBE	0000 0000 _H
–	Reserved	0000 11AC _H	SV,U	nBE	0000 0000 _H
ERAY_NMV1	Network Management Vector 1	0000 11B0 _H	SV,U	nBE	0000 0000 _H
ERAY_NMV2	Network Management Vector 2	0000 11B4 _H	SV,U	nBE	0000 0000 _H
ERAY_NMV3	Network Management Vector 3	0000 11B8 _H	SV,U	nBE	0000 0000 _H
–	Reserved	0000 11BC _H - 0000 12FC _H	nBE	nBE	0000 0000 _H

Message Buffer Control Registers

ERAY_MRC	Message RAM Configuration	0000 1300 _H	SV,U	SV,U	0180 0000 _H
ERAY_FRF	FIFO Rejection Filter	0000 1304 _H	SV,U	SV,U	0180 0000 _H
ERAY_FRFM	FIFO Rejection Filter Mask	0000 1308 _H	SV,U	SV,U	0000 0000 _H
ERAY_FCL	FIFO Critical Level	0000 130C _H	SV,U	SV,U	0000 0080 _H

Message Buffer Status Registers

ERAY_MHDS	Message Handler Status	0000 1310 _H	SV,U	SV,U	0000 0080 _H
ERAY_LDTS	Last Dynamic Transmit Slot	0000 1314 _H	SV,U	SV,U	0000 0000 _H
ERAY_FSR	FIFO Status Register	0000 1318 _H	SV,U	SV,U	0000 0000 _H

Register Description

Table 11-2 Address Map of FlexRay Communication Controller (E-Ray) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
FlexRay Communication Controller (E-Ray)					
ERAY_MHDF	Message Handler Constraints Flags	0000 131C _H	SV,U	nBE	0000 0000 _H
ERAY_TXRQ1	Transmission Request 1	0000 1320 _H	SV,U	nBE	0000 0000 _H
ERAY_TXRQ2	Transmission Request 2	0000 1324 _H	SV,U	nBE	0000 0000 _H
ERAY_TXRQ3	Transmission Request 3	0000 1328 _H	SV,U	nBE	0000 0000 _H
ERAY_TXRQ4	Transmission Request 4	0000 132C _H	SV,U	nBE	0000 0000 _H
ERAY_NDAT1	New Data 1	0000 1330 _H	SV,U	nBE	0000 0000 _H
ERAY_NDAT2	New Data 2	0000 1334 _H	SV,U	nBE	0000 0000 _H
ERAY_NDAT3	New Data 3	0000 1338 _H	SV,U	nBE	0000 0000 _H
ERAY_NDAT4	New Data 4	0000 133C _H	SV,U	nBE	0000 0000 _H
ERAY_MBSC1	Message Buffer Status Changed 1	0000 1340 _H	SV,U	nBE	0000 0000 _H
ERAY_MBSC2	Message Buffer Status Changed 2	0000 1344 _H	SV,U	nBE	0000 0000 _H
ERAY_MBSC3	Message Buffer Status Changed 3	0000 1348 _H	SV,U	nBE	0000 0000 _H
ERAY_MBSC4	Message Buffer Status Changed 4	0000 134C _H	SV,U	nBE	0000 0000 _H
–	Reserved	0000 1350 _H - 0000 13EC _H	SV,U	nBE	0000 0000 _H

Register Description
Table 11-2 Address Map of FlexRay Communication Controller (E-Ray) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	

FlexRay Communication Controller (E-Ray)
Identification Register

ERAY_CREL	Core Release Registers	0000 13F0 _H	SV,U	nBE	xxxx xxxx _H
ERAY_ENDN	Endian Register	0000 13F4 _H	SV,U	nBE	8765 4321 _H
–	Reserved	0000 13F8 _H - 0000 13FC _H	SV,U	nBE	0000 0000 _H

Input Buffer

ERAY_WRDS01	Write Data Section 01	0000 1400 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS02	Write Data Section 02	0000 1404 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS03	Write Data Section 03	0000 1408 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS04	Write Data Section 04	0000 140C _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS05	Write Data Section 05	0000 1410 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS06	Write Data Section 06	0000 1414 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS07	Write Data Section 07	0000 1418 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS08	Write Data Section 08	0000 141C _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS09	Write Data Section 09	0000 1420 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS10	Write Data Section 10	0000 1424 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS11	Write Data Section 11	0000 1428 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS12	Write Data Section 12	0000 142C _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS13	Write Data Section 13	0000 1430 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS14	Write Data Section 14	0000 1434 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS15	Write Data Section 15	0000 1438 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS16	Write Data Section 16	0000 143C _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS17	Write Data Section 17	0000 1440 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS18	Write Data Section 18	0000 1444 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS19	Write Data Section 19	0000 1448 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS20	Write Data Section 20	0000 144C _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS21	Write Data Section 21	0000 1450 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS22	Write Data Section 22	0000 1454 _H	SV,U	SV,U	0000 0000 _H

Register Description
Table 11-2 Address Map of FlexRay Communication Controller (E-Ray) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
FlexRay Communication Controller (E-Ray)					
ERAY_WRDS23	Write Data Section 23	0000 1458 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS24	Write Data Section 24	0000 145C _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS25	Write Data Section 25	0000 1460 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS26	Write Data Section 26	0000 1464 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS27	Write Data Section 27	0000 1468 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS28	Write Data Section 28	0000 146C _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS29	Write Data Section 29	0000 1470 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS30	Write Data Section 30	0000 1474 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS31	Write Data Section 31	0000 1478 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS32	Write Data Section 32	0000 147C _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS33	Write Data Section 33	0000 1480 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS34	Write Data Section 34	0000 1484 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS35	Write Data Section 35	0000 1488 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS36	Write Data Section 36	0000 148C _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS37	Write Data Section 37	0000 1490 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS38	Write Data Section 38	0000 1494 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS39	Write Data Section 39	0000 1498 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS40	Write Data Section 40	0000 149C _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS41	Write Data Section 41	0000 14A0 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS42	Write Data Section 42	0000 14A4 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS43	Write Data Section 43	0000 14A8 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS44	Write Data Section 44	0000 14AC _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS45	Write Data Section 45	0000 14B0 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS46	Write Data Section 46	0000 14B4 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS47	Write Data Section 47	0000 14B8 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS48	Write Data Section 48	0000 14BC _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS49	Write Data Section 49	0000 14C0 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRDS50	Write Data Section 50	0000 14C4	SV,U	SV,U	0000 0000 _H

Register Description
Table 11-2 Address Map of FlexRay Communication Controller (E-Ray) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
FlexRay Communication Controller (E-Ray)					
ERAY_WRDS51	Write Data Section 51	0000 14C8	SV,U	SV,U	0000 0000 _H
ERAY_WRDS52	Write Data Section 52	0000 14CC	SV,U	SV,U	0000 0000 _H
ERAY_WRDS53	Write Data Section 53	0000 14D0	SV,U	SV,U	0000 0000 _H
ERAY_WRDS54	Write Data Section 54	0000 14D4	SV,U	SV,U	0000 0000 _H
ERAY_WRDS55	Write Data Section 55	0000 14D8	SV,U	SV,U	0000 0000 _H
ERAY_WRDS56	Write Data Section 56	0000 14DC	SV,U	SV,U	0000 0000 _H
ERAY_WRDS57	Write Data Section 57	0000 14E0	SV,U	SV,U	0000 0000 _H
ERAY_WRDS58	Write Data Section 58	0000 14E4	SV,U	SV,U	0000 0000 _H
ERAY_WRDS59	Write Data Section 59	0000 14E8	SV,U	SV,U	0000 0000 _H
ERAY_WRDS60	Write Data Section 60	0000 14EC	SV,U	SV,U	0000 0000 _H
ERAY_WRDS61	Write Data Section 61	0000 14F0	SV,U	SV,U	0000 0000 _H
ERAY_WRDS62	Write Data Section 63	0000 14F4	SV,U	SV,U	0000 0000 _H
ERAY_WRDS63	Write Data Section 63	0000 14F8	SV,U	SV,U	0000 0000 _H
ERAY_WRDS64	Write Data Section 64	0000 14FC	SV,U	SV,U	0000 0000 _H
ERAY_WRHS1	Write Header Section 1	0000 1500 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRHS2	Write Header Section 2	0000 1504 _H	SV,U	SV,U	0000 0000 _H
ERAY_WRHS3	Write Header Section 3	0000 1508 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 150C _H	nBE	nBE	0000 0000 _H
ERAY_IBCM	Input Buffer Command Mask	0000 1510 _H	SV,U	SV,U	0000 0000 _H
ERAY_IBCR	Input Buffer Command Request	0000 1514 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 1518 _H - 0000 15FC _H	nBE	nBE	0000 0000 _H
Output Buffer					
ERAY_RDDS01	Read Data Section 01	0000 1600 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS02	Read Data Section 02	0000 1604 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS03	Read Data Section 03	0000 1608 _H	SV,U	nBE	0000 0000 _H

Register Description

Table 11-2 Address Map of FlexRay Communication Controller (E-Ray) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
FlexRay Communication Controller (E-Ray)					
ERAY_RDDS04	Read Data Section 04	0000 160C _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS05	Read Data Section 05	0000 1610 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS06	Read Data Section 06	0000 1614 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS07	Read Data Section 07	0000 1618 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS08	Read Data Section 08	0000 161C _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS09	Read Data Section 09	0000 1620 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS10	Read Data Section 10	0000 1624 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS11	Read Data Section 11	0000 1628 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS12	Read Data Section 12	0000 162C _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS13	Read Data Section 13	0000 1630 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS14	Read Data Section 14	0000 1634 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS15	Read Data Section 15	0000 1638 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS16	Read Data Section 16	0000 163C _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS17	Read Data Section 17	0000 1640 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS18	Read Data Section 18	0000 1644 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS19	Read Data Section 19	0000 1648 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS20	Read Data Section 20	0000 164C _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS21	Read Data Section 21	0000 1650 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS22	Read Data Section 22	0000 1654 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS23	Read Data Section 23	0000 1658 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS24	Read Data Section 24	0000 165C _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS25	Read Data Section 25	0000 1660 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS26	Read Data Section 26	0000 1664 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS27	Read Data Section 27	0000 1668 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS28	Read Data Section 28	0000 166C _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS29	Read Data Section 29	0000 1670 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS30	Read Data Section 30	0000 1674 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS31	Read Data Section 31	0000 1678 _H	SV,U	nBE	0000 0000 _H

Register Description
Table 11-2 Address Map of FlexRay Communication Controller (E-Ray) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
FlexRay Communication Controller (E-Ray)					
ERAY_RDDS32	Read Data Section 32	0000 167C _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS33	Read Data Section 33	0000 1680 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS34	Read Data Section 34	0000 1684 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS35	Read Data Section 35	0000 1688 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS36	Read Data Section 36	0000 168C _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS37	Read Data Section 37	0000 1690 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS38	Read Data Section 38	0000 1694 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS39	Read Data Section 39	0000 1698 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS40	Read Data Section 40	0000 169C _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS41	Read Data Section 41	0000 16A0 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS42	Read Data Section 42	0000 16A4 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS43	Read Data Section 43	0000 16A8 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS44	Read Data Section 44	0000 16AC _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS45	Read Data Section 45	0000 16B0 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS46	Read Data Section 46	0000 16B4 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS47	Read Data Section 47	0000 16B8 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS48	Read Data Section 48	0000 16BC _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS49	Read Data Section 49	0000 16C0 _H	SV,U	nBE	0000 0000 _H
ERAY_RDDS50	Read Data Section 50	0000 16C4	SV,U	nBE	0000 0000 _H
ERAY_RDDS51	Read Data Section 51	0000 16C8	SV,U	nBE	0000 0000 _H
ERAY_RDDS52	Read Data Section 52	0000 16CC	SV,U	nBE	0000 0000 _H
ERAY_RDDS53	Read Data Section 53	0000 16D0	SV,U	nBE	0000 0000 _H
ERAY_RDDS54	Read Data Section 54	0000 16D4	SV,U	nBE	0000 0000 _H
ERAY_RDDS55	Read Data Section 55	0000 16D8	SV,U	nBE	0000 0000 _H
ERAY_RDDS56	Read Data Section 56	0000 16DC	SV,U	nBE	0000 0000 _H
ERAY_RDDS57	Read Data Section 57	0000 16E0	SV,U	nBE	0000 0000 _H
ERAY_RDDS58	Read Data Section 58	0000 16E4	SV,U	nBE	0000 0000 _H
ERAY_RDDS59	Read Data Section 59	0000 16E8	SV,U	nBE	0000 0000 _H

Register Description

Table 11-2 Address Map of FlexRay Communication Controller (E-Ray) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
FlexRay Communication Controller (E-Ray)					
ERAY_RDDS60	Read Data Section 60	0000 16EC	SV,U	nBE	0000 0000 _H
ERAY_RDDS61	Read Data Section 61	0000 16F0	SV,U	nBE	0000 0000 _H
ERAY_RDDS62	Read Data Section 63	0000 16F4	SV,U	nBE	0000 0000 _H
ERAY_RDDS63	Read Data Section 63	0000 16F8	SV,U	nBE	0000 0000 _H
ERAY_RDDS64	Read Data Section 64	0000 16FC	SV,U	nBE	0000 0000 _H
ERAY_RDHS1	Read Header Section 1	0000 1700 _H	SV,U	nBE	0000 0000 _H
ERAY_RDHS2	Read Header Section 2	0000 1704 _H	SV,U	nBE	0000 0000 _H
ERAY_RDHS3	Read Header Section 3	0000 1708 _H	SV,U	nBE	0000 0000 _H
ERAY_MBS	Message Buffer Status	0000 170C _H	SV,U	nBE	0000 0000 _H
ERAY_OBCM	Output Buffer Command Mask	0000 1710 _H	SV,U	SV,U	0000 0000 _H
ERAY_OBCR	Output Buffer Command Request	0000 1714 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 1718 _H - 0000 17F8 _H	nBE	nBE	0000 0000 _H
–	Reserved	0000 17FC _H	nBE ²⁾	nBE	0000 0000 _H

1) All accesses of the SAK-CIC310-OSMX2HT, regardless of the host interface in use, are executed in Supervisor Mode.

2) BE if accessing via SSC host interface using autoincrement mode (transaction header: INCE=1).

11.1.2 SCU Kernel Registers

Table 11-3 shows all registers associated with the SCU.

Table 11-3 SCU Registers

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
SCU Registers					
SCU_OSCCON	Oscillator Control Register	0000 0800 _H	SV,U	SV,U	0000 000C _H
—	Reserved	0000 0804 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0808 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 080C _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0810 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0814 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0818 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 081C _H	nBE	nBE	0000 0000 _H
SCU_SYSCON	System and PLL Control Register	0000 0820 _H	SV,U	SV,U	0817 020F _H
—	Reserved	0000 0824 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0828 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 082C _H	nBE	nBE	0000 0000 _H
SCU_CHTR0	Channel 0 Trigger Register	0000 0830 _H	SV,U	SV,U	0000 0000 _H
SCU_CHTR1	Channel 1 Trigger Register	0000 0834 _H	SV,U	SV,U	0000 0000 _H
SCU_CHTR2	Channel 2 Trigger Register	0000 0838 _H	SV,U	SV,U	0000 0000 _H
SCU_CHTR3	Channel 3 Trigger Register	0000 083C _H	SV,U	SV,U	0000 0000 _H
SCU_CHTR4	Channel 4 Trigger Register	0000 0840 _H	SV,U	SV,U	0000 0000 _H
SCU_CHTR5	Channel 5 Trigger Register	0000 0844 _H	SV,U	SV,U	0000 0000 _H

Register Description
Table 11-3 SCU Registers (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
SCU Registers					
SCU_CHTR6	Channel 6 Trigger Register	0000 0848 _H	SV,U	SV,U	0000 0000 _H
SCU_CHTR7	Channel 7 Trigger Register	0000 084C _H	SV,U	SV,U	0000 0000 _H
SCU_ETCTR	External Trigger Control Register	0000 0850 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 0854 _H	nBE	nBE	0000 0000 _H
SCU_SRCR	Service Request Control Register	0000 0858 _H	SV,U	SV,U	0000 0007 _H
SSC_ERRCUM	SSC Cumulative Error Register	0000 085C _H	SV,U	SV,U	0000 0000 _H
SCU_IDCHIP	Chip Identification Register	0000 0860 _H	SV,U	BE	0000 2104 _H
SCU_IDCHIP	Chip Identification Register	0000 0860 _H	SV,U	BE	0000 2103 _H
XMU_CON	XMU Global Control Register	0000 0864 _H	SV,U	SV,U	0000 0000 _H
SCU_GSBIR	Gather Scattered Bits Input Register	0000 0868 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 086C _H	nBE	nBE	0000 0000 _H
SCU_GSBMR	Gather Scattered Bits Mask Register	0000 0870 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 0874 _H	nBE	nBE	0000 0000 _H
SCU_GSBOR	Gather Scattered Bits Output Register	0000 0878 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 087C _H	nBE	nBE	0000 0000 _H
SCU_DMADAT00	DMA Data 00 Register	0000 0880 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT01	DMA Data 01 Register	0000 0884 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT02	DMA Data 02 Register	0000 0888 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT03	DMA Data 03 Register	0000 088C _H	SV,U	SV,U	0000 0000 _H

Register Description
Table 11-3 SCU Registers (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
SCU Registers					
SCU_DMADAT04	DMA Data 04 Register	0000 0890 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT05	DMA Data 05 Register	0000 0894 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT06	DMA Data 06 Register	0000 0898 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT07	DMA Data 07 Register	0000 089C _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT08	DMA Data 08 Register	0000 08A0 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT09	DMA Data 09 Register	0000 08A4 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT10	DMA Data 10 Register	0000 08A8 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT11	DMA Data 11 Register	0000 08AC _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT12	DMA Data 12 Register	0000 08B0 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT13	DMA Data 13 Register	0000 08B4 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT14	DMA Data 14 Register	0000 08B8 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT15	DMA Data 15 Register	0000 08BC _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT16	DMA Data 16 Register	0000 08C0 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT17	DMA Data 17 Register	0000 08C4 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT18	DMA Data 18 Register	0000 08C8 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT19	DMA Data 19 Register	0000 08CC _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT20	DMA Data 20 Register	0000 08D0 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT21	DMA Data 21 Register	0000 08D4 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT22	DMA Data 22 Register	0000 08D8 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT23	DMA Data 23 Register	0000 08DC _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT24	DMA Data 24 Register	0000 08E0 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT25	DMA Data 25 Register	0000 08E4 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT26	DMA Data 26 Register	0000 08E8 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT27	DMA Data 27 Register	0000 08EC _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT28	DMA Data 28 Register	0000 08F0 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT29	DMA Data 29 Register	0000 08F4 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT30	DMA Data 30 Register	0000 08F8 _H	SV,U	SV,U	0000 0000 _H
SCU_DMADAT31	DMA Data 31 Register	0000 08FC _H	SV,U	SV,U	0000 0000 _H

Register Description

- 1) All accesses of the SAK-CIC310-OSMX2HT, regardless of the host interface in use, are executed in Supervisor Mode.

11.1.3 DMA Registers

Table 11-4 shows all registers associated with the DMA Kernel.

Table 11-4 Address Map of Direct Memory Access Controller (DMA)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Direct Memory Access Controller (DMA)					
–	Reserved	0000 0400 _H	nBE	nBE	0000 0000 _H
–	Reserved	0000 0404 _H	nBE	nBE	0000 0000 _H
DMA_ID	DMA Module Identification Register	0000 0408 _H	SV,U ₂₎	BE	001A C012 _H
–	Reserved	0000 040C _H	BE	BE	0000 0000 _H
DMA_CHRSTR	DMA Channel Reset Request Register	0000 0410 _H	SV,U	SV	0000 0000 _H
DMA_TRSR	DMA Transaction Request State Register	0000 0414 _H	SV,U	BE	0000 0000 _H
DMA_STREQ	DMA Software Transaction Request Register	0000 0418 _H	SV,U	SV	0000 0000 _H
DMA_HTREQ	DMA Hardware Transaction Request Register	0000 041C _H	SV,U	SV	0000 0000 _H
DMA_EER	DMA Enable Error Register	0000 0420 _H	SV,U	SV	0000 0000 _H
DMA_ERRSR	DMA Error Status Register	0000 0424 _H	SV,U	BE	0000 0000 _H
DMA_CLRE	DMA Clear Error Register	0000 0428 _H	SV,U	SV	0000 0000 _H
DMA_GINTR	DMA Global Interrupt Set Register	0000 042C _H	SV,U	SV	0000 0000 _H
DMA_MESR	DMA Move Engine Status Register	0000 0430 _H	SV,U	BE	0000 0000 _H
DMA_ME0R	DMA Move Engine 0 Read Register	0000 0434 _H	SV,U ₂₎	BE	0000 0000 _H
–	Reserved	0000 0438 _H	BE	BE	0000 0000 _H

Register Description
Table 11-4 Address Map of Direct Memory Access Controller (DMA) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Direct Memory Access Controller (DMA)					
DMA_ME0PR	DMA Move Engine 0 Pattern Register	0000 043C _H	SV,U ²⁾	SV	0000 0000 _H
—	Reserved	0000 0440 _H	BE	BE	0000 0000 _H
—	Reserved	0000 0444 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0448 _H	nBE ²⁾	nBE	0000 0000 _H
—	Reserved	0000 044C _H	BE	BE	0000 0000 _H
—	Reserved	0000 0450 _H	BE	BE	0000 0000 _H
DMA_INTSR	DMA Interrupt Status Register	0000 0454 _H	SV,U	BE	0000 0000 _H
DMA_INTCR	DMA Interrupt Clear Register	0000 0458 _H	SV,U	SV	0000 0000 _H
DMA_WRPSR	DMA Wrap Status Register	0000 045C _H	SV,U ²⁾	BE	0000 0000 _H
—	Reserved	0000 0460 _H	BE	BE	0000 0000 _H
—	Reserved	0000 0464 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0468 _H	nBE ²⁾	nBE	0000 0000 _H
—	Reserved	0000 046C _H - 0000 047C _H	BE	BE	0000 0000 _H
DMA_CHSR00	DMA Channel 00 Status Register	0000 0480 _H	SV,U	BE	0000 0000 _H
DMA_CHCR00	DMA Channel 00 Control Register	0000 0484 _H	SV,U	SV	0000 0000 _H
DMA_CHICR00	DMA Channel 00 Interrupt Control Register	0000 0488 _H	SV,U	SV	0000 0000 _H
DMA_ADRCR00	DMA Channel 00 Address Control Register	0000 048C _H	SV,U	SV	0000 0000 _H
DMA_SADR00	DMA Channel 00 Source Address Register	0000 0490 _H	SV,U	SV	0000 0000 _H

Register Description

Table 11-4 Address Map of Direct Memory Access Controller (DMA) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Direct Memory Access Controller (DMA)					
DMA_DADR00	DMA Channel 00 Destination Address Register	0000 0494 _H	SV,U	SV	0000 0000 _H
DMA_SHADR00	DMA Channel 00 Shadowed Address Register	0000 0498 _H	SV,U ²⁾	BE	0000 0000 _H
–	Reserved	0000 049C _H	BE	BE	0000 0000 _H
DMA_CHSR01	DMA Channel 01 Status Register	0000 04A0 _H	SV,U	BE	0000 0000 _H
DMA_CHCR01	DMA Channel 01 Control Register	0000 04A4 _H	SV,U	SV	0000 0000 _H
DMA_CHICR01	DMA Channel 01 Interrupt Control Register	0000 04A8 _H	SV,U	SV	0000 0000 _H
DMA_ADRCR01	DMA Channel 01 Address Control Register	0000 04AC _H	SV,U	SV	0000 0000 _H
DMA_SADR01	DMA Channel 01 Source Address Register	0000 04B0 _H	SV,U	SV	0000 0000 _H
DMA_DADR01	DMA Channel 01 Destination Address Register	0000 04B4 _H	SV,U	SV	0000 0000 _H
DMA_SHADR01	DMA Channel 01 Shadowed Address Register	0000 04B8 _H	SV,U ²⁾	BE	0000 0000 _H
–	Reserved	0000 04BC _H	BE	BE	0000 0000 _H
DMA_CHSR02	DMA Channel 02 Status Register	0000 04C0 _H	SV,U	BE	0000 0000 _H
DMA_CHCR02	DMA Channel 02 Control Register	0000 04C4 _H	SV,U	SV	0000 0000 _H
DMA_CHICR02	DMA Channel 02 Interrupt Control Register	0000 04C8 _H	SV,U	SV	0000 0000 _H
DMA_ADRCR02	DMA Channel 02 Address Control Register	0000 04CC _H	SV,U	SV	0000 0000 _H

Register Description
Table 11-4 Address Map of Direct Memory Access Controller (DMA) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Direct Memory Access Controller (DMA)					
DMA_SADR02	DMA Channel 02 Source Address Register	0000 04D0 _H	SV,U	SV	0000 0000 _H
DMA_DADR02	DMA Channel 02 Destination Address Register	0000 04D4 _H	SV,U	SV	0000 0000 _H
DMA_SHADR02	DMA Channel 02 Shadowed Address Register	0000 04D8 _H	SV,U ²⁾	BE	0000 0000 _H
–	Reserved	0000 04DC _H	BE	BE	0000 0000 _H
DMA_CHSR03	DMA Channel 03 Status Register	0000 04E0 _H	SV,U	BE	0000 0000 _H
DMA_CHCR03	DMA Channel 03 Control Register	0000 04E4 _H	SV,U	SV	0000 0000 _H
DMA_CHICR03	DMA Channel 03 Interrupt Control Register	0000 04E8 _H	SV,U	SV	0000 0000 _H
DMA_ADRCR03	DMA Channel 03 Address Control Register	0000 04EC _H	SV,U	SV	0000 0000 _H
DMA_SADR03	DMA Channel 03 Source Address Register	0000 04F0 _H	SV,U	SV	0000 0000 _H
DMA_DADR03	DMA Channel 03 Destination Address Register	0000 04F4 _H	SV,U	SV	0000 0000 _H
DMA_SHADR03	DMA Channel 03 Shadowed Address Register	0000 04F8 _H	SV,U ²⁾	BE	0000 0000 _H
–	Reserved	0000 04FC _H	BE	BE	0000 0000 _H
DMA_CHSR04	DMA Channel 04 Status Register	0000 0500 _H	SV,U	BE	0000 0000 _H
DMA_CHCR04	DMA Channel 04 Control Register	0000 0504 _H	SV,U	SV	0000 0000 _H
DMA_CHICR04	DMA Channel 04 Interrupt Control Register	0000 0508 _H	SV,U	SV	0000 0000 _H

Register Description

Table 11-4 Address Map of Direct Memory Access Controller (DMA) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Direct Memory Access Controller (DMA)					
DMA_ADRCR04	DMA Channel 04 Address Control Register	0000 050C _H	SV,U	SV	0000 0000 _H
DMA_SADR04	DMA Channel 04 Source Address Register	0000 0510 _H	SV,U	SV	0000 0000 _H
DMA_DADR04	DMA Channel 04 Destination Address Register	0000 0514 _H	SV,U	SV	0000 0000 _H
DMA_SHADR04	DMA Channel 04 Shadowed Address Register	0000 0518 _H	SV,U ²⁾	BE	0000 0000 _H
—	Reserved	0000 051C _H	BE	BE	0000 0000 _H
DMA_CHSR05	DMA Channel 05 Status Register	0000 0520 _H	SV,U	BE	0000 0000 _H
DMA_CHCR05	DMA Channel 05 Control Register	0000 0524 _H	SV,U	SV	0000 0000 _H
DMA_CHICR05	DMA Channel 05 Interrupt Control Register	0000 0528 _H	SV,U	SV	0000 0000 _H
DMA_ADRCR05	DMA Channel 05 Address Control Register	0000 052C _H	SV,U	SV	0000 0000 _H
DMA_SADR05	DMA Channel 05 Source Address Register	0000 0530 _H	SV,U	SV	0000 0000 _H
DMA_DADR05	DMA Channel 05 Destination Address Register	0000 0534 _H	SV,U	SV	0000 0000 _H
DMA_SHADR05	DMA Channel 05 Shadowed Address Register	0000 0538 _H	SV,U ²⁾	BE	0000 0000 _H
—	Reserved	0000 053C _H	BE	BE	0000 0000 _H
DMA_CHSR06	DMA Channel 06 Status Register	0000 0540 _H	SV,U	BE	0000 0000 _H
DMA_CHCR06	DMA Channel 06 Control Register	0000 0544 _H	SV,U	SV	0000 0000 _H

Register Description

Table 11-4 Address Map of Direct Memory Access Controller (DMA) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Direct Memory Access Controller (DMA)					
DMA_CHICR06	DMA Channel 06 Interrupt Control Register	0000 0548 _H	SV,U	SV	0000 0000 _H
DMA_ADRCR06	DMA Channel 06 Address Control Register	0000 054C _H	SV,U	SV	0000 0000 _H
DMA_SADR06	DMA Channel 06 Source Address Register	0000 0550 _H	SV,U	SV	0000 0000 _H
DMA_DADR06	DMA Channel 06 Destination Address Register	0000 0554 _H	SV,U	SV	0000 0000 _H
DMA_SHADR06	DMA Channel 06 Shadowed Address Register	0000 0558 _H	SV,U ²⁾	BE	0000 0000 _H
–	Reserved	0000 055C _H	BE	BE	0000 0000 _H
DMA_CHSR07	DMA Channel 07 Status Register	0000 0560 _H	SV,U	BE	0000 0000 _H
DMA_CHCR07	DMA Channel 07 Control Register	0000 0564 _H	SV,U	SV	0000 0000 _H
DMA_CHICR07	DMA Channel 07 Interrupt Control Register	0000 0568 _H	SV,U	SV	0000 0000 _H
DMA_ADRCR07	DMA Channel 07 Address Control Register	0000 056C _H	SV,U	SV	0000 0000 _H
DMA_SADR07	DMA Channel 07 Source Address Register	0000 0570 _H	SV,U	SV	0000 0000 _H
DMA_DADR07	DMA Channel 07 Destination Address Register	0000 0574 _H	SV,U	SV	0000 0000 _H
DMA_SHADR07	DMA Channel 07 Shadowed Address Register	0000 0578 _H	SV,U ²⁾	BE	0000 0000 _H
–	Reserved	0000 057C _H	BE	BE	0000 0000 _H
–	Reserved	0000 0580 _H	nBE ²⁾	nBE	0000 0000 _H

Register Description

Table 11-4 Address Map of Direct Memory Access Controller (DMA) (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Direct Memory Access Controller (DMA)					
—	Reserved	0000 0584 _H - 0000 0588 _H	BE	BE	0000 0000 _H
—	Reserved	0000 058C _H - 0000 05A8 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 05AC _H	nBE ²⁾	nBE	0000 0000 _H
—	Reserved	0000 05B0 _H - 0000 05B4 _H	BE	BE	0000 0000 _H
—	Reserved	0000 05B8 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 05BC _H	nBE ²⁾	nBE	0000 0000 _H
—	Reserved	0000 05C0 _H - 0000 05EC _H	BE	BE	0000 0000 _H
—	Reserved	0000 05F0 _H - 0000 05F8 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 05FC _H	nBE ²⁾	nBE	0000 0000 _H
—	Reserved	0000 0600 _H - 0000 06FC _H	BE	BE	0000 0000 _H

1) All accesses of the SAK-CIC310-OSMX2HT, regardless of the host interface in use, are executed in Supervisor Mode.

2) BE if accessing via SSC host interface using auto increment mode (transaction header: INCE=1).

11.1.4 MLI Kernel Registers

Table 11-5 shows all registers associated with the MLI Kernel.

Table 11-5 MLI Kernel Registers

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Micro Link Interface (MLI)					
–	Reserved	0000 0200 _H	nBE	nBE	0000 0000 _H
–	Reserved	0000 0204 _H	nBE	nBE	0000 0000 _H
MLI_ID	Module Identification Register	0000 0208 _H	SV,U	BE	0025 C006 _H
MLI_FDR	Fractional Divider Register	0000 020C _H	SV,U	SV,U	03FF 43FF _H
MLI_TCR	Transmitter Control Register	0000 0210 _H	SV,U	SV,U	0000 0110 _H
MLI_TSTATR	Transmitter Status Register	0000 0214 _H	SV,U	BE	0000 0000 _H
MLI_TP0STATR	Transmitter Pipe 0 Status Register	0000 0218 _H	SV,U	BE	0000 0000 _H
MLI_TP1STATR	Transmitter Pipe 1 Status Register	0000 021C _H	SV,U	BE	0000 0000 _H
MLI_TP2STATR	Transmitter Pipe 2 Status Register	0000 0220 _H	SV,U	BE	0000 0000 _H
MLI_TP3STATR	Transmitter Pipe 3 Status Register	0000 0224 _H	SV,U	BE	0000 0000 _H
MLI_TCMDR	Transmitter Command Register	0000 0228 _H	SV,U	SV,U	0000 0000 _H
MLI_TRSTATR	Transmitter Registers Status Register	0000 022C _H	SV,U	BE	0000 0000 _H
MLI_TP0AOFR	Transmitter Pipe 0 Address Offset Register	0000 0230 _H	SV,U	BE	0000 0000 _H
MLI_TP1AOFR	Transmitter Pipe 1 Address Offset Register	0000 0234 _H	SV,U	BE	0000 0000 _H
MLI_TP2AOFR	Transmitter Pipe 2 Address Offset Register	0000 0238 _H	SV,U	BE	0000 0000 _H
MLI_TP3AOFR	Transmitter Pipe 3 Address Offset Register	0000 023C _H	SV,U	BE	0000 0000 _H

Register Description
Table 11-5 MLI Kernel Registers (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Micro Link Interface (MLI)					
MLI_TP0DATAR	Transmitter Pipe 0 Data Register	0000 0240 _H	SV,U	BE	0000 0000 _H
MLI_TP1DATAR	Transmitter Pipe 1 Data Register	0000 0244 _H	SV,U	BE	0000 0000 _H
MLI_TP2DATAR	Transmitter Pipe 2 Data Register	0000 0248 _H	SV,U	BE	0000 0000 _H
MLI_TP3DATAR	Transmitter Pipe 3 Data Register	0000 024C _H	SV,U	BE	0000 0000 _H
MLI_TDRAR	Transmitter Data Read Answer Register	0000 0250 _H	SV,U	SV,U	0000 0000 _H
MLI_TP0BAR	Transmitter Pipe 0 Base Address Register	0000 0254 _H	SV,U	SV,U	0000 0000 _H
MLI_TP1BAR	Transmitter Pipe 1 Base Address Register	0000 0258 _H	SV,U	SV,U	0000 0000 _H
MLI_TP2BAR	Transmitter Pipe 2 Base Address Register	0000 025C _H	SV,U	SV,U	0000 0000 _H
MLI_TP3BAR	Transmitter Pipe 3 Base Address Register	0000 0260 _H	SV,U	SV,U	0000 0000 _H
MLI_TCBAR	Transmitter Copy Base Address Register	0000 0264 _H	SV,U	BE	0000 0000 _H
MLI_RCR	Receiver Control Register	0000 0268 _H	SV,U	SV,U	0000 0000 _H
MLI_RP0BAR	Receiver Pipe 0 Base Address Register	0000 026C _H	SV,U	BE	0000 0000 _H
MLI_RP1BAR	Receiver Pipe 1 Base Address Register	0000 0270 _H	SV,U	BE	0000 0000 _H
MLI_RP2BAR	Receiver Pipe 2 Base Address Register	0000 0274 _H	SV,U	BE	0000 0000 _H
MLI_RP3BAR	Receiver Pipe 3 Base Address Register	0000 0278 _H	SV,U	BE	0000 0000 _H
MLI_RP0STATR	Receiver Pipe 0 Status Register	0000 027C _H	SV,U	BE	0000 0000 _H

Register Description
Table 11-5 MLI Kernel Registers (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Micro Link Interface (MLI)					
MLI_RP1STATR	Receiver Pipe 1 Status Register	0000 0280 _H	SV,U	BE	0000 0000 _H
MLI_RP2STATR	Receiver Pipe 2 Status Register	0000 0284 _H	SV,U	BE	0000 0000 _H
MLI_RP3STATR	Receiver Pipe 3 Status Register	0000 0288 _H	SV,U	BE	0000 0000 _H
MLI_RADRR	Receiver Address Register	0000 028C _H	SV,U	BE	0000 0000 _H
MLI_RDATAR	Receiver Data Register	0000 0290 _H	SV,U	BE	0000 0000 _H
MLI_SCR	Set Clear Register	0000 0294 _H	SV,U	SV,U	0000 0010 _H
MLI_TIER	Transmitter Interrupt Enable Register	0000 0298 _H	SV,U	SV,U	0000 0000 _H
MLI_TISR	Transmitter Interrupt Status Register	0000 029C _H	SV,U	BE	0000 0000 _H
MLI_TINPR	Transmitter Interrupt Node Pointer Register	0000 02A0 _H	SV,U	SV,U	0000 0000 _H
MLI_RIER	Receiver Interrupt Enable Register	0000 02A4 _H	SV,U	SV,U	0000 0000 _H
MLI_RISR	Receiver Interrupt Status Register	0000 02A8 _H	SV,U	BE	0000 0000 _H
MLI_RINPR	Receiver Interrupt Node Pointer Register	0000 02AC _H	SV,U	SV,U	0000 0000 _H
MLI_GINTR	Global Interrupt Set Register	0000 02B0 _H	SV,U	SV,U	0000 0000 _H
MLI_OICR	Output Input Control Register	0000 02B4 _H	SV,U	SV,U	1000 8000 _H
—	Reserved	0000 02B8 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 02BC _H	nBE ²⁾	nBE	0000 0000 _H

1) All accesses of the SAK-CIC310-OSMX2HT, regardless of the host interface in use, are executed in Supervisor Mode.

2) BE if accessing via SSC host interface using autoincrement mode (transaction header: INCE=1).

11.1.5 SSC Kernel Registers

Table 11-6 shows all registers associated with the SSC Kernel.

Table 11-6 SSC Kernel Registers

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Synchronous Serial Interface (SSC)					
–	Reserved	0000 0900 _H	nBE	nBE	0000 0000 _H
SSC_PISEL	Port Input Select Register	0000 0904 _H	SV,U	SV,U	0000 0000 _H
SSC_ID	Module Identification Register	0000 0908 _H	SV,U	nBE	0000 4510 _H
–	Reserved	0000 090C _H	nBE	nBE	0000 0000 _H
SSC_CON	Control Register	0000 0910 _H	SV,U	SV,U	0000 0000 _H
SSC_BR	Baud Rate Timer Reload Register	0000 0914 _H	SV,U	SV,U	0000 0000 _H
SSC_SSOC	Slave Select Output Control Register	0000 0918 _H	SV,U	SV,U	0000 0000 _H
SSC_SSOTC	Slave Select Output Timing Control Register	0000 091C _H	SV,U	SV,U	0000 0000 _H
SSC_TB	Transmit Buffer Register	0000 0920 _H	SV,U	SV,U	0000 0000 _H
SSC_RB	Receive Buffer Register	0000 0924 _H	SV,U	SV,U	0000 0000 _H
SSC_STAT	Status Register	0000 0928 _H	SV,U	SV,U	0000 0000 _H
SSC_EFM	Error Flag Modification Register	0000 092C _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 0930 _H - 0000 09FC _H	SV,U	nBE	0000 0000 _H
SSC_ERRCUM	SSC Cumulative Error Register	0000 085C _H	SV,U	SV,U	0000 0000 _H

1) All accesses of the SAK-CIC310-OSMX2HT, regardless of the host interface in use, are executed in Supervisor Mode.

Register Description

11.1.6 XMU Kernel Registers

Table 11-7 shows all registers associated with the XMU.

Table 11-7 XMU Registers

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	

XMU Registers

XMU_CON	XMU Global Control Register	0000 0864 _H	SV,U	SV,U	0000 0000 _H
---------	-----------------------------	------------------------	------	------	------------------------

1) All accesses of the SAK-CIC310-OSMX2HT, regardless of the host interface in use, are executed in Supervisor Mode.

11.1.7 Ports Registers

Table 11-8 shows all registers associated with the Ports.

Table 11-8 Port Registers

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	

Port Registers

Port 0

P0_OUT	Port 0 Output Register	0000 0A00 _H	SV,U	SV,U	0000 0000 _H
P0_OMR	Port 0 Output Modification Register	0000 0A04 _H	SV,U	SV,U	0000 0000 _H
–	Reserved	0000 0A08 _H	nBE	nBE	0000 0000 _H
–	Reserved	0000 0A0C _H	nBE	nBE	0000 0000 _H
P0_IOCRO	Port 0 Input/Output Control Register 0	0000 0A10 _H	SV,U	SV,U	2020 9090 _H
P0_IOCRO4	Port 0 Input/Output Control Register 4	0000 0A14 _H	SV,U	SV,U	2020 2020 _H
P0_IOCRO8	Port 0 Input/Output Control Register 8	0000 0A18 _H	SV,U	SV,U	2020 2020 _H
P0_IOCRO12	Port 0 Input/Output Control Register 12	0000 0A1C _H	SV,U	SV,U	0000 2220 _H
–	Reserved	0000 0A20 _H	nBE	nBE	0000 0000 _H

Register Description
Table 11-8 Port Registers (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	

Port Registers

P0_IN	Port 0 Input Register	0000 0A24 _H	SV,U	nBE	0000 XXXX _H
P0_PDR	Port 0 Pad Driver Mode Register	0000 0A28 _H	SV,U	SV,U	0000 5500 _H
—	Reserved	0000 0A2C _H	nBE	nBE	0000 0000 _H

Port 1

P1_OUT	Port 1 Output Register	0000 0A30 _H	SV,U	SV,U	0000 0000 _H
P1_OMR	Port 1 Output Modification Register	0000 0A34 _H	SV,U	SV,U	0000 0000 _H
—	Reserved	0000 0A38 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0A3C _H	nBE	nBE	0000 0000 _H
P1_IOCRO	Port 1 Input/Output Control Register 0	0000 0A40 _H	SV,U	SV,U	2222 2222 _H
P1_IOCRA4	Port 1 Input/Output Control Register 4	0000 0A44 _H	SV,U	SV,U	2222 2222 _H
P1_IOCRA8	Port 1 Input/Output Control Register 8	0000 0A48 _H	SV,U	SV,U	2222 2222 _H
P1_IOCRA12	Port 1 Input/Output Control Register 12	0000 0A4C _H	SV,U	SV,U	2222 2222 _H
—	Reserved	0000 0A50 _H	nBE	nBE	0000 0000 _H
P1_IN	Port 1 Input Register	0000 0A54 _H	SV,U	nBE	0000 XXXX _H
P1_PDR	Port 1 Pad Driver Mode Register	0000 0A58 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0A5C _H	nBE	nBE	0000 0000 _H

Port 2

—	Reserved	0000 0A60 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0A64 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0A68 _H	nBE	nBE	0000 0000 _H
—	Reserved	0000 0A6C _H	nBE	nBE	0000 0000 _H
P2_IOCRO	Port 2 Input/Output Control Register 0	0000 0A70 _H	SV,U	SV,U	2020 2020 _H

Register Description

Table 11-8 Port Registers (cont'd)

Short Name	Description	Address	Access Mode ¹⁾		Reset Value
			Read	Write	
Port Registers					
P2_IOCRR4	Port 2 Input/Output Control Register 4	0000 0A74 _H	SV,U	SV,U	2020 2020 _H
P2_IOCRR8	Port 2 Input/Output Control Register 8	0000 0A78 _H	SV,U	SV,U	2020 2020 _H
P2_IOCRR12	Port 2 Input/Output Control Register 8	0000 0A7C _H	SV,U	SV,U	2020 2020 _H
–	Reserved	0000 0A80 _H	nBE	nBE	0000 0000 _H
P2_IN	Port 2 Input Register	0000 0A84 _H	SV,U	nBE	0000 XXXX _H
–	Reserved	0000 0A88 _H - 0000 0AFC _H	nBE	nBE	0000 0000 _H

1) All accesses of the SAK-CIC310-OSMX2HT, regardless of the host interface in use, are executed in Supervisor Mode.

11.1.8 Reserved Address Ranges not Associated to Peripheral

Table 11-7 shows all address ranges not associated with any peripheral.

Table 11-9 Reserved Address Ranges

Short Name	Description	Address	Access Mode		Reset Value
			Read	Write	
Address Ranges not Associated with a Peripheral					
–	Reserved	0000 0000 _H - 0000 01FC _H	BE	BE	0000 0000 _H
–	Reserved	0000 02C0 _H - 0000 03FC _H	BE	BE	0000 0000 _H
–	Reserved	0000 0700 _H - 0000 07FC _H	BE	BE	0000 0000 _H
–	Reserved	0000 0B00 _H - 0000 0FFC _H	nBE	nBE	0000 0000 _H
–	Reserved	0000 1800 _H - 0000 FFFC _H	BE	BE	0000 0000 _H

Keyword Index

This section lists a number of keywords which refer to specific details of the SAK-CIC310-OSMX2HT in terms of its architecture, its functional units, or functions. Bold page number entries identify the main definition material for a topic.

A

Abbreviations 1-6
Access mode definitions 1-5

B

baud rate
SSC 4-40

C

Clock System 3-5
Oscillator run detection 3-9
Clock system
CGU block diagram 3-13
Module clock generation
Fractional divider 7-61
OSCCON 3-11
Oscillator Run Detection 3-10
Oscillator run detection 3-9
PLL Lock Detection 3-22
PLL, see "PLL"
SYSCON 3-24
Control
reset 3-4

D

DMA 5-1
Address update 5-17
Block diagram 1-15, 5-4
Channel request 5-8
Channel reset 5-16
Circular buffer 5-13
Definition of terms 5-2
Interrupt generation 5-14
Operation 5-4
Operation modes 5-8
Principle 5-3

Registers

Overview **5-22**

Request assignment 5-20
Shadow registers 5-5
Transaction control 5-19

Document

Structure 1-1
Terminology and abbreviations 1-3
Textual conventions 1-1

E

E-Ray

Channel protocol controller 2-3
Clock
Minimum 2-223
Communication controller
FIFO 2-209
Known bugs 2-241
Message RAM 2-228
Controller access 2-227
Data partition 2-233
Data transfer 2-217
Data transfer to output RAM
2-215
Header partition 2-230
Host access 2-211
Parity 2-234
Worst case timing 2-221
POC 2-100
Protocol operation control 2-100
Restrictions 2-240
Service requests 2-237
Customer host interface 2-2
Frame processing 2-4
Generic host interface 2-3
Global time unit 2-4

Input buffer 2-3	EIER 2-51
Interrupt control 2-5	EIES 2-46
Kernel	EILS 2-38
Clock minimum 2-223	EIR 2-28
RAM test 2-25	ENDN 2-150
Release coding 2-149	ESIDnn (nn = 01-015) 2-117
Service request 2-28–2-71	FCM 2-127
Test registers 2-18	FRF 2-124
Kernel block diagram 2-2	FRFM 2-126
Message buffer pointer 2-123	FSR 2-132
Message handler 2-3	GTUC01 2-87
Message RAM 2-3	GTUC02 2-88
Module registers 2-6	GTUC03 2-89
Network management 2-4	GTUC04 2-90
Output Buffer 2-3	GTUC05 2-91
Protocol	GTUC06 2-92
Clock 2-179	GTUC07 2-93
Global time 2-179	GTUC08 2-94
Local time 2-179	GTUC09 2-95
Synchronization 2-180	GTUC10 2-96
Communication controller	GTUC11 2-97
Commands 2-76–2-78	IBCM 2-157
States 2-185–2-201	IBCR 2-159
Communication cycle 2-176	ID 2-12
Configuration 2-177	Identification 2-148–2-150
Dynamic segment 2-177	ILE 2-66
Network idle time 2-177	Input buffer 2-150–2-159
Static segment 2-176	LCK 2-26
Symbol window 2-177	LDS 2-131
Network management 2-201	MBS 2-168
Register offsets 2-6	MBSC1 2-144
Registers 2-6–2-11	MBSC2 2-145
ACS 2-114	MBSC3 2-146
CCEV 2-104	MBSC4 2-147
CCSV 2-99	Message buffer control 2-121–2-127
Communication controller control 2-71–2-97	Message buffer status 2-128–2-147
Communication controller status 2-99–2-120	MHDC 2-86
CREL 2-148	MHDF 2-134
CUST1 2-13	MHDS 2-128
CUST3 2-15	MRC 2-121
Customer registers 2-11	MTCCV 2-106

NDAT1 **2-140**
 NDAT2 **2-141**
 NDAT3 **2-142**
 NDAT4 **2-143**
 NEMC **2-82**
 NMVnn (nn = 1-3) **2-120**
 OBCM **2-173**
 OBCR **2-174**
 OCV **2-108**
 OSIDnn (nn = 01-15) **2-118**
 Output buffer 2-161–2-174
 PRT2 **2-85**
 PRTC1 **2-83**
 RCV **2-107**
 RDDSnn (nn = 01-64) **2-161**
 RDHS1 **2-162**
 RDHS2 **2-164**
 RDHS3 **2-166**
 SCV **2-105**
 Service request 2-28–2-71
 SFS **2-109**
 SIER **2-61**
 SIES **2-56**
 SILS **2-42**
 SIR **2-33**
 STPW1 **2-69**
 STPW2 **2-71**
 SUCC1 **2-72**
 SUCC2 **2-80**
 SUCC3 **2-81**
 SWINIT **2-111**
 T1C **2-68**
 Test registers 2-18
 TEST1 **2-18**
 TEST2 **2-23**
 TOC **2-67**
 TXRQ1 **2-136**
 TXRQ2 **2-137**
 TXRQ3 **2-138**
 TXRQ4 **2-139**
 WRDSnn (nn = 01-64) **2-151**
 WRHS **2-152**
 WRHS2 **2-155**

WRHS3 **2-156**
 Symbol processing 2-4
 System universal control 2-4
 Transient buffer 2-3
 External bus interface unit
 Example configuration 8-2
 External to internal operation 8-5
 Access control 8-9
 Address extension diagram 8-8
 Address translation 8-6
 Basic timing 8-10, 8-11
 Signal direction 8-5
 Features 1-14, 8-1
 Overview 8-1
 Registers 8-13
 Overview 8-13
 Signal description 8-3
 External Memory Interface Unit 8-1–8-16
 Basic operation 8-2
 Block diagram 8-1
 Registers
 XMU_CON **3-47, 8-13**
 XMU_EXTCON **8-19**

F

Fractional divider 7-61
 Block diagram 7-61
 Operating modes 7-62

M

MLI

Applications 7-1
 Communication principles 7-5
 General description 7-6
 Handshake timing 7-13
 Interrupts 7-47
 Naming conventions 7-4
 Reading process 7-46
 Receiver 7-34
 Error handling 7-42
 I/O control 7-44
 Operation modes 7-34
 Registers

GINTR **7-103**
 OICR **7-104**
 Overview 7-64
 RADDR **7-89**
 RDATAR **7-90**
 RIER **7-98**
 RINPR **7-101**
 RISR **7-100**
 RPxBAR **7-87**
 SCR **7-91**
 TCBAR **7-83**
 TDRAR **7-81**
 TIER **7-93**
 TINPR **7-96**
 TISR **7-95**
 TPxAOFR **7-79**
 TPxBAR **7-82**
 TPxDATAR **7-80**
 Startup 7-15
 Timings 7-50
 Transaction flow diagrams
 Copy base address 7-54
 Transmitter 7-17
 Errors 7-31
 I/O control 7-33
 Operation modes 7-17
 Parity 7-31
 Transfer mode selection 7-28
 Transmission format 7-20
 Transmission modes 7-21

P

Package outline 1-20
 Pin definitions and functions 1-21–1-31
 pinning
 pin description **1-19**
 PLL **3-13**
 Features 3-14
 Functionality 3-15
 Loss-of-lock 3-27
 Port
 Registers
 P0_IN **9-24**

P0_IOCRO **9-18**
 P0_IOCRL2 **9-23**
 P0_IOCRL4 **9-20**
 P0_IOCRL8 **9-22**
 P0_OMR **9-17**
 P0_OUT **9-16**
 P0_PDR 9-25
 P1_IN **9-39**
 P1_IOCRO **9-33**, 9-33
 P1_IOCRL2 **9-37**
 P1_IOCRL4 **9-35**
 P1_IOCRL8 **9-36**
 P1_OMR **9-32**
 P1_OUT **9-31**
 P1_PDR 9-40
 P2_IN **9-52**
 P2_IOCRO **9-47**
 P2_IOCRL2 **9-51**
 P2_IOCRL4 **9-49**
 P2_IOCRL8 **9-50**
 Pn_IN **9-7**
 Pn_IOCRO **9-8**
 Pn_IOCRL2 **9-11**
 Pn_IOCRL4 9-9, **9-9**
 Pn_IOCRL8 **9-10**
 Pn_OMR **9-6**
 Pn_OUT **9-5**
 Pn_PDR 9-12

Port 0 **9-13**

Port 1 9-29

Port 2 9-45

Ports

Driver characteristics selection 9-3, 9-25
 Input register Pn_IN **9-7**
 Input/output control registers
 Pn_IOCRLx **9-8**
 Output modification register Pn_OMR **9-6**
 Output register Pn_OUT **9-5**
 Pad driver control 9-2
 Port 0 **9-13**
 I/O functions 9-26, 9-41, 9-53

Port 1 9-29
Port 2 9-45
Port control coding 9-4

R

Reset
control block 3-4

S

SCU
Registers
OSCCON **3-11**

SSC
Baud rate generation 6-10, 6-16
Block diagram 6-3
Chip select generation 6-13
communication protocol **4-43**
Error detection 6-16
format error 4-57, 6-35, 6-36, 6-38
Full-duplex operation 6-5
Half-duplex operation 6-7
Interrupts 6-16
register read access **4-45**
register write access **4-51**
Registers 6-19–6-33
BR **6-32**
Offset addresses 3-1, 6-19
Overview 3-1, 6-19
PISEL **6-20**
RB **6-34**
TB **6-33**
slave select **4-43**
Slave select input operation 6-12

STCU

Registers
DMADATnn **3-50**

X

XMU
Registers
CON **3-47, 8-13**
XMU, see "External Memory Interface
Unit"

Register Index

This section lists the references to the Registers of the SAK-CIC310-OSMX2HT.

D

DMA controller registers 22

DMA_ADRCR00 19

DMA_ADRCR01 20

DMA_ADRCR02 20

DMA_ADRCR03 21

DMA_ADRCR04 22

DMA_ADRCR05 22

DMA_ADRCR06 23

DMA_ADRCR07 23

DMA_ADRCR0n 52

DMA_CHCR00 19

DMA_CHCR01 20

DMA_CHCR02 20

DMA_CHCR03 21

DMA_CHCR04 21

DMA_CHCR05 22

DMA_CHCR06 22

DMA_CHCR07 23

DMA_CHCR0n 45

DMA_CHICR00 19

DMA_CHICR01 20

DMA_CHICR02 20

DMA_CHICR03 21

DMA_CHICR04 21

DMA_CHICR05 22

DMA_CHICR06 23

DMA_CHICR07 23

DMA_CHICR0n (n = 0-7) 50

DMA_CHRSTR 27, 18

DMA_CHSR00 19

DMA_CHSR01 20

DMA_CHSR02 20

DMA_CHSR03 21

DMA_CHSR04 21

DMA_CHSR05 22

DMA_CHSR06 22

DMA_CHSR07 23

DMA_CHSR0n 49

DMA_CLRE 18

DMA_DADR00 20

DMA_DADR01 20

DMA_DADR02 21

DMA_DADR03 21

DMA_DADR04 22

DMA_DADR05 22

DMA_DADR06 23

DMA_DADR07 23

DMA_DADR0n 56

DMA_EER 31, 18

DMA_ERRSR 33, 18

DMA_GINTR 18

DMA_HTREQ 18

DMA_ID 25, 18

DMA_INTCR 19

DMA_INTSR 19

DMA_ME0PR 43, 19

DMA_ME0R 18

DMA_MESR 40, 18

DMA_SADR00 19

DMA_SADR01 20

DMA_SADR02 21

DMA_SADR03 21

DMA_SADR04 22

DMA_SADR05 22

DMA_SADR06 23

DMA_SADR07 23

DMA_SADR0n 55

DMA_SHADR00 20

DMA_SHADR01 20

DMA_SHADR02 21

DMA_SHADR03 21

DMA_SHADR04 22

DMA_SHADR05 22

DMA_SHADR06 23
 DMA_SHADR07 23
 DMA_SHADR0n 57
 DMA_STREQ 18
 DMA_TRSR 28, 18
 DMA_WRPSR 19

E

ERAY_ACS 8, 114, 5
 ERAY_CCEV 8, 104, 4
 ERAY_CCSV 8, 99, 4
 ERAY_CREL 10, 148
 ERAY_CUST0 6, 11, 2
 ERAY_CUST1 7, 13, 2
 ERAY_CUST3 7, 15, 2
 ERAY_EIER 7, 51, 2
 ERAY_EIES 7, 46, 2
 ERAY_EILS 7, 38, 2
 ERAY_EIR 7, 28, 2
 ERAY_ENDN 10, 150
 ERAY_ESIDnn (nn = 01-15) 9, 117
 ERAY_ESIDnn (nn=01-15) 5–6
 ERAY_FCL 9
 ERAY_FCM 127
 ERAY_FRF 9, 124, 6
 ERAY_FRFM 9, 126, 6
 ERAY_FSR 9, 132
 ERAY_GTUC01 8, 87
 ERAY_GTUC02 8, 88
 ERAY_GTUC03 8, 89
 ERAY_GTUC04 8, 90
 ERAY_GTUC05 8, 91
 ERAY_GTUC06 8, 92
 ERAY_GTUC07 8, 93
 ERAY_GTUC08 8, 94
 ERAY_GTUC09 8, 95
 ERAY_GTUC10 8, 96
 ERAY_GTUC11 8, 97
 ERAY_GTUCnn (nn=01-11) 3–4
 ERAY_IBCM 10, 157, 10
 ERAY_IBCR 10, 159, 10
 ERAY_ID 7, 12, 2
 ERAY_ILE 7, 66, 3

ERAY_LCK 7, 26, 2
 ERAY_LDTS 9, 131
 ERAY_MBS 10, 168, 13
 ERAY_MBSC1 9, 144, 7
 ERAY_MBSC2 9, 145, 7
 ERAY_MBSC3 10, 146, 7
 ERAY_MBSC4 10, 147, 7
 ERAY_MHDC 8, 86
 ERAY_MHDC1 3
 ERAY_MHDF 9, 134
 ERAY_MHDS 9, 128, 6
 ERAY_MRC 9, 121, 6
 ERAY_MTCCV 8, 106, 4
 ERAY_NDAT1 9, 140, 7
 ERAY_NDAT2 9, 141, 7
 ERAY_NDAT3 9, 142, 7
 ERAY_NDAT4 9, 143, 7
 ERAY_NEMC 7, 82, 3
 ERAY_NMV1 6
 ERAY_NMV2 6
 ERAY_NMV3 6
 ERAY_NMVn (n = 1-3) 9, 120
 ERAY_OBCM 10, 173, 13
 ERAY_OBCR 10, 174, 13
 ERAY_OCV 8, 108, 4
 ERAY_OSIDnn (nn = 01-15) 9, 118
 ERAY_OSIDnn (nn=01-15) 5
 ERAY_PRTC1 7, 83, 3
 ERAY_PRTC2 8, 85, 3
 ERAY_RCV 8, 107, 4
 ERAY_RDDSn (nn = 01-64) 10, 161
 ERAY_RDDSn (nn=01-64) 10–13
 ERAY_RDHS1 10, 162, 13
 ERAY_RDHS2 10, 164, 13
 ERAY_RDHS3 10, 166, 13
 ERAY_SCV 8, 105, 4
 ERAY_SFS 8, 109, 4
 ERAY_SIER 7, 61, 3
 ERAY_SIES 7, 56, 2
 ERAY_SILS 42, 2
 ERAY_SIR 7, 33, 2
 ERAY_STPW 3
 ERAY_STPW1 7, 69

ERAY_STPW2 7, 71
 ERAY_SUCC1 7, 72, 3
 ERAY_SUCC2 7, 80, 3
 ERAY_SUCC3 7, 81, 3
 ERAY_SWINIT 111
 ERAY_SWNIT 8, 5
 ERAY_T0C 7, 67, 3
 ERAY_T1C 7, 68, 3
 ERAY_TEST1 7, 18, 2
 ERAY_TEST2 7, 23, 2
 ERAY_TXRQ1 9, 136, 7
 ERAY_TXRQ2 9, 137, 7
 ERAY_TXRQ3 9, 138, 7
 ERAY_TXRQ4 9, 139, 7
 ERAY_WRDSnn (nn = 01-64) 10, 151
 ERAY_WRDSnn (nn=01-64) 8–10
 ERAY_WRHS1 10, 152, 10
 ERAY_WRHS2 10, 155, 10
 ERAY_WRHS3 10, 156, 10

M

MLI module registers 25
 MLI_FDR 110, 25
 MLI_GINTR 103
 MLI_ID 67, 25
 MLI_OICR 104
 MLI_RADDR 89
 MLI_RCR 84
 MLI_RDATAR 90
 MLI_RIER 98
 MLI_RINPR 101
 MLI_RISR 100
 MLI_RPnBAR (n = 0-3) 87
 MLI_RPnSTATR (n = 0-3) 88
 MLI_SCR 91
 MLI_TCBAR 83
 MLI_TCMR 75
 MLI_TCR 68, 3, 25
 MLI_TDRAR 81
 MLI_TIER 93
 MLI_TINPR 96
 MLI_TISR 95
 MLI_TP0STATR 25

MLI_TPnBAR (n = 0-3) 82
 MLI_TPnDATAR (n = 0-3) 80
 MLI_TPxAOFR 79
 MLI_TPxSTATR 73
 MLI_TRSTATR 77
 MLI_TSTATR 71, 25

P

P0_IN 30
 P0_IOCR0 29
 P0_IOC12 29
 P0_IOC4 29
 P0_IOC8 29
 P0_OMR 29
 P0_OUT 29
 P0_PDR 25, 30
 P1_IN 30
 P1_IOCR0 30
 P1_IOC12 30
 P1_IOC4 30
 P1_IOC8 30
 P1_OMR 30
 P1_OUT 30
 P1_PDR 40
 P2_IN 31
 P2_IOCR0 30
 P2_IOC12 31
 P2_IOC4 31
 P2_IOC8 31
 Pn_PDR 12
 Port module registers 29

S

SCU controller registers 1
 SCU_CHTR0 14
 SCU_CHTR1 14
 SCU_CHTR2 14
 SCU_CHTR3 14
 SCU_CHTR4 14
 SCU_CHTR5 14
 SCU_CHTR6 15
 SCU_CHTR7 15
 SCU_DMADATnn (nn=00-31) 15–16

SCU_ERRCUM 15, 28
SCU_ETCTR 15
SCU_IDCHIP 15
SCU_OSCCON 14
SCU_SRCR 15
SSC module registers 1, 19
SSC_BR 32, 28
SSC_CON 23, 28
SSC_EFM 27, 28
SSC_ID 22, 28
SSC_PISEL 28
SSC_RB 34, 28
SSC_SSOC 29, 28
SSC_SSOTC 30, 28
SSC_STAT 25, 28
SSC_TB 33, 28
SSC0_STAT 31, 39
SSC1_STAT 31, 39

X

XMU module registers 13
XMU_CON 47, 13, 15, 29

<http://www.infineon.com>